



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**THE ROLE OF EFFICIENT XML INTERCHANGE (EXI)
IN NAVY WIDE-AREA NETWORK (WAN)
OPTIMIZATION**

by

Steven J. Debich

March 2015

Thesis Advisor:

Don Brutzman

Co-Advisor:

Scot Miller

Second Reader:

Don McGregor

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2015	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE THE ROLE OF EFFICIENT XML INTERCHANGE (EXI) IN NAVY WIDE-AREA NETWORK (WAN) OPTIMIZATION			5. FUNDING NUMBERS	
6. AUTHOR Debich, Steven J.			8. PERFORMING ORGANIZATION REPORT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME AND ADDRESS Commander Navy Information Dominance Forces (COMNAVIDFOR) 115 Lake View Parkway Suffolk, VA 23435			11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.	
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT Navy afloat units become disadvantaged users, once disconnected from the pier, due in part to the high latency associated with SATCOM. Unfortunately recent gains in SATCOM capacity alone do not overcome throughput limitations that result from latency's effect on connection-oriented protocols. To mitigate the effect of latency and other performance inhibiting factors, the Navy is improving its current WAN optimization capabilities by implementing Riverbed Steelhead WOCs. At-sea testing has shown Steelhead increases effective SATCOM capacity by 50%. Laboratory testing demonstrates that by encoding structured and semi-structured data as EXI rather than XML, compression ratios can be further improved, up to 19 times greater than Steelhead's compression capability alone. Combining EXI with Steelhead will further improve the efficient use of existing SATCOM capacity and enable greater operational capabilities, when operating in a communications constrained environment. Not only does EXI improve compactness of traffic traveling over relatively high capacity SATCOM channels, it also expands net-centric capabilities to devices operating at the edge of the network that are restricted to lower capacity transmission methods. In order to achieve these substantial improvements the Navy must incorporate the already mandated DISR standard, EXI, as the single standard for all systems transferring structured and semi-structured data.				
14. SUBJECT TERMS EXI, efficient xml interchange, EFX, efficient xml, Riverbed, Steelhead, WAN optimization, compression, long fat network, LFN			15. NUMBER OF PAGES 163	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**THE ROLE OF EFFICIENT XML INTERCHANGE (EXI) IN NAVY WIDE-
AREA NETWORK (WAN) OPTIMIZATION**

Steven J. Debich
Lieutenant Commander, United States Navy
B.S., The Pennsylvania State University, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN NETWORK OPERATIONS AND TECHNOLOGY

from the

**NAVAL POSTGRADUATE SCHOOL
March 2015**

Author: Steven J. Debich

Approved by: Don Brutzman
Thesis Advisor

Scot Miller
Co-Advisor

Don McGregor
Second Reader

Dan Boger
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Navy afloat units become disadvantaged users, once disconnected from the pier, due in part to the high latency associated with SATCOM. Unfortunately recent gains in SATCOM capacity alone do not overcome throughput limitations that result from latency's effect on connection-oriented protocols. To mitigate the effect of latency and other performance inhibiting factors, the Navy is improving its current WAN optimization capabilities by implementing Riverbed Steelhead WOCs.

At-sea testing has shown Steelhead increases effective SATCOM capacity by 50%. Laboratory testing demonstrates that by encoding structured and semi-structured data as EXI rather than XML, compression ratios can be further improved, up to 19 times greater than Steelhead's compression capability alone. Combining EXI with Steelhead will further improve the efficient use of existing SATCOM capacity and enable greater operational capabilities, when operating in a communications constrained environment.

Not only does EXI improve compactness of traffic traveling over relatively high capacity SATCOM channels, it also expands net-centric capabilities to devices operating at the edge of the network that are restricted to lower capacity transmission methods. In order to achieve these substantial improvements the Navy must incorporate the already mandated DISR standard, EXI, as the single standard for all systems transferring structured and semi-structured data.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	PROBLEM STATEMENT	2
B.	MOTIVATION AND PURPOSE	3
C.	RESEARCH QUESTIONS	4
D.	HYPOTHESIS.....	5
E.	METHODOLOGY	5
F.	BENEFITS OF STUDY.....	7
G.	THESIS ORGANIZATION.....	8
II.	RELATED WORK.....	9
A.	LAN VERSUS WAN.....	10
B.	SATCOM.....	11
C.	NETWORK TRAFFIC OBSTACLES	11
1.	Network and Transport.....	11
2.	Application and Protocol.....	14
3.	Operating System and Hardware.....	15
D.	WAN OPTIMIZATION TECHNIQUES	16
1.	Protocol Optimization	16
2.	Caching	17
3.	Prefetching.....	17
4.	Compression and Data Deduplication	18
E.	WAN OPTIMIZATION SOLUTIONS	18
F.	COMPRESSION.....	19
1.	General Compression and Data Deduplication.....	20
2.	Format Specific Compression	21
3.	Compression Limitations	22
G.	XML COMPRESSION	24
1.	Verbose by Design.....	24
2.	Generic Compression Approaches	25
3.	Binary Encoding Approaches	25
4.	EXI	27
H.	CHAPTER SUMMARY.....	30
III.	METHODS	31
A.	DATA SET SELECTION	31
B.	NETWORK CONFIGURATION	33
C.	DATA MEASUREMENT AND TRANSFER METHODS	35
D.	FEATURES TESTED	42
1.	LZ Compression.....	42
2.	Adaptive Compression.....	45
3.	SDR.....	46
4.	LZ Compression and SDR	49
E.	CHAPTER SUMMARY.....	49

IV.	EXPERIMENTAL RESULTS AND ANALYSIS.....	51
A.	INTERPRETING THE RESULTS.....	51
B.	COMPRESSION—RESEARCH QUESTIONS 1 AND 2	54
C.	ADAPTIVE COMPRESSION—RESEARCH QUESTION 3	59
D.	COMPRESSION + SDR COLD TRANSFER—RESEARCH QUESTION 4	61
E.	COMPRESSION + SDR WARM TRANSFER—RESEARCH QUESTION 5	63
F.	SDR EFFECT ON A SINGLE MODIFICATION AT BEGINNING, MIDDLE, END—RESEARCH QUESTIONS 6 AND 7	66
G.	SDR EFFECT ON GREATER THAN ONE MODIFICATION— RESEARCH QUESTIONS 8 AND 9	70
H.	CHAPTER SUMMARY.....	71
V.	CONCLUSIONS AND RECOMMENDATIONS.....	73
A.	THE ROLE OF EXI IN NAVY WAN OPTIMIZATION	73
1.	Limitations Imposed by Encryption.....	73
2.	Diminished Returns Due to Redundant Compression	74
B.	ESTABLISHING AN AUDITABLE BANDWIDTH BUDGET	75
C.	ESTABLISHING AND ACCELERATING EXI ADOPTION IN THE NAVY	77
D.	FINAL THOUGHTS	78
	APPENDIX A. DATA SETS AND SCHEMA LOCATIONS.....	79
	APPENDIX B. PROCEEDINGS ARTICLE.....	81
	APPENDIX C. RIVERBED FLEET INSTALLATION MESSAGE	85
	APPENDIX D. PRESENTATION PREPARED FOR NAVIDFOR.....	91
	LIST OF REFERENCES.....	137
	INITIAL DISTRIBUTION LIST	143

LIST OF FIGURES

Figure 1.	Simplified network diagram depicting ship and shore components of EXI host-based, and Steelhead network-based optimization.	6
Figure 2.	Passing disk-based compression data segments via reference number (from Bentrup et al., 2012, p. 14).	21
Figure 3.	Comparison between the OSI and TCP/IP networking models. Host-based format specific compression is generally applied at the application layer. Network-based general compression is commonly applied at the Network and Transport layers. (after J. Green, personal communication, 12 December 2014).	22
Figure 4.	WAP system diagram showing XML transformation into WML via a gateway architecture (after Saha, Jamtgaard, & Villasenor, 2001).	26
Figure 5.	SPAWAR System Center Pacific—Network Technology and Integration Laboratory (NTIL) equipment string.	34
Figure 6.	Riverbed Steelhead Current Connections Report, expanded view.	36
Figure 7.	Establishing a persistent connection using <i>tcpforward</i>	37
Figure 8.	Sending a file using <i>tcpsend</i>	38
Figure 9.	Successful file transfer reflected by <i>tcpforward</i>	38
Figure 10.	Current connections screen used to identify 118 bytes of overhead for connections using neither LZ Compression or SDR.	40
Figure 11.	Current connections screen used to identify 147 bytes of overhead for connections using LZ Compression only.	40
Figure 12.	Current connections screen used to identify 147 bytes of overhead for connections using SDR only.	41
Figure 13.	Current connections screen used to identify 148 bytes of overhead for connections using LZ Compression and SDR combined.	41
Figure 14.	LZ Compression in-path rule configuration.	43
Figure 15.	Enabling LZ Compression in-path rule.	44
Figure 16.	Disabling Adaptive Compression.	44
Figure 17.	Enabling Adaptive Compression.	45
Figure 18.	SDR Performance Configuration Options.	47
Figure 19.	SDR in-path rule configuration.	47
Figure 20.	Enabling SDR in-path rule.	48
Figure 21.	Graph showing compression ratio comparisons for FSM, JRAE and VMF data sets. Larger Y-axis values indicate greater compression.	55
Figure 22.	Graph showing additional compression factor gained by using EXI over SH Comp for FSM, JRAE, and VMF data sets. The values shown indicate by how many times better EXI compression is than SH Comp.	55
Figure 23.	Graph comparing Steelhead compression levels 1,6, and 9 for the FSM data set.	57
Figure 24.	Graph comparing Steelhead compression levels 1,6, and 9 for the JRAE data set.	58

Figure 25.	Graph comparing Steelhead compression levels 1,6, and 9 for the VMF data set.	58
Figure 26.	Graph showing SDR warm compression ratio comparisons for FSM, JRAE and VMF data sets. Larger Y-axis values indicate greater compression.	64
Figure 27.	Graph showing additional compression factor gained by using EXI over XML for FSM, JRAE, and VMF data sets. The values shown indicate by how many times better EXI compression is than SH Comp.....	65
Figure 28.	Graph for FSM file “Planning Transaction Dtl” with modifications occurring at the beginning, middle, and end of the file. Files transferred individually.	68
Figure 29.	Graph for FSM file “Planning Transaction Dtl” with modifications occurring at the beginning, middle, and end of the file. Files transferred in succession.	68
Figure 30.	EXI performance improvement over XML for SDR warm transfers containing a modification at the beginning, middle, and end of the file.	69

LIST OF TABLES

Table 1.	Minimum requirements for a binary XML format as determined by XBC Working Group (after Goldman & Lenkov, 2005).....	28
Table 2.	Example calculations comparing compression ratio and space savings calculations. Results can be misleading if not carefully considered.....	52
Table 3.	Terms and abbreviations used for graph legends and analysis.	53
Table 4.	Data table listing original file size and resulting file size following SH Comp and EXI for FSM, JRAE, and VMF data sets.	56
Table 5.	Data table listing compression ratios achieved using SH Comp, EXI, and corresponding improvement factors for FSM, JRAE, and VMF data sets.	56
Table 6.	Data table showing compression ratios achieved using Steelhead LZ Compression levels 1, 6, and 9 for FSM, JRAE, and VMF data sets.	59
Table 7.	Data table listing original file size, resulting file size, and the difference following the application of Steelhead Adaptive Compression to Gzip and EXI data for FSM, JRAE, and VMF data sets.	60
Table 8.	Raw data table showing the change in compression ratios for Gzip and EXI after the application of Steelhead Adaptive Compression.	61
Table 9.	Data table listing original file size, resulting cold transfer file size, and the difference following the application of Steelhead's SDR to previously compressed data for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.	62
Table 10.	Data table listing original compression ratio and resulting SDR cold transfer compression ratio for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.	63
Table 11.	Data table listing original file size, resulting SDR cold and warm transfer file sizes for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.	65
Table 12.	Data table listing original compression ratio and resulting SDR cold and warm transfer compression ratio for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.	66
Table 13.	Data table showing original file size and payload sizes for FSM file "Planning Transaction Dtl" with modifications occurring at the beginning, middle, and end of the file. Includes data for files transferred individually and in succession.....	69
Table 14.	Data table showing compression ratios for FSM file "Planning Transaction Dtl" with modifications occurring at the beginning, middle, and end of the file. Includes ratios for files transferred individually and in succession. Additionally an improvement factor is shown illustrating negligible difference in overall compression when adding SH Comp to EXI.....	70

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

16PSK	16-Phase Shift Keying
ADNS	Automated Digital Network System
AFATDS	Advanced Field Artillery Tactical Data System
BAC	Baseline Allowance Control
BDP	Bandwidth Delay Product
BFTN	Battle Force Tactical Network
BPSK	Binary-Phase Shift Keying
C2	Command and Control
C2PC	Command and Control Personal Computer
CANES	Consolidated Afloat Networks and Enterprise Services
CCE	Communications Contested Environment
CIFS	Common Internet File System
CNA	Center for Naval Analysis
CPU	Central Processing Unit
CRIME	Compression Ratio Info-Leak Made Easy
DOD	Department of Defense
DSCS	Defense Satellite Communications System
EFX	Efficient XML
EMCON	Emission Control
EXI	Efficient XML Interchange
FAM	Functional Area Manager
FI	Fast Infoset
FIFO	First-In, First-Out
FSM	Food Service Management
GBS	Global Broadcast Service
GCCS-M	Global Command and Control System-Maritime
HTTP	Hypertext Transfer Protocol
IA	Information Assurance
IANA	Internet Assigned Numbers Authority
ID	Information Dominance

ISP	Internet Service Provider
JTM	Joint Target Management
JRAE 06	Joint Rapid Architecture Experiment 2006
JEFX 06	Joint Expeditionary Force Experiment 2006
LAN	Local Area Network
LFN	Long Fat Network
LRU	Least Recently Used
LZ	Lempel-Ziv
LZ77	Lempel-Ziv 1977
MAPI	Messaging Application Programming Interface
MILSATCOM	Military SATCOM
MUOS	Mobile User Objective System
NAVSUP	Navy Supply Systems Command
NIAPS	Navy Information Application Product Suite
NIC	Network Interface Card
NOC	Network Operations Center
NTIL	Network Technology and Integration Laboratory
OSD	Office of the Secretary of Defense
OSI	Open Systems Interconnection
PDF	Portable Document Format
PII	Personally Identifiable Information
QoS	Quality of Service
RTT	Round Trip Time
SaaS	Software as a Service
SATCOM	Satellite Communications
SBIR	Small Business Innovation Research
SDR	Scalable Data Referencing
SGML	Standard Generalized Markup Language
SOA	Service-Oriented Architecture
SPAWAR	Space and Naval Warfare Systems Command
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol

TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
USFF	U.S. Fleet Forces
VMF	Variable Message Format
WAN	Wide-Area Network
WAP	Wireless Application Protocol
WBXML	WAP Binary XML
WGS	Wideband Global SATCOM; Wideband Gapfiller Satellite
WML	Wireless Markup Language
WOC	WAN Optimization Controller
XBC	XML Binary Characterization
XML	Extensible Markup Language

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and most importantly, I would like to thank my wife, Amanda, for her love and support. I thank you for your willingness to proofread my often-dismal rough drafts and listen to my lengthy explanations of various network optimization methods, in which I'm certain you still have no interest. Your encouragement, patience, and selflessness are only a few of the reasons why you are truly my better half.

I am grateful for the exceptional support provided by John Schneider at AgileDelta. The near-real-time email responses you provided to my endless number of questions pertaining to EXI and testing methodologies ensured that I never wandered too far off course. I appreciate your willingness to extend your time and knowledge to a master's student you had never met.

Thank you to all of the SPAWAR, NAVSUP, and other Navy systems commands personnel who assisted me in learning about the large and complex systems and organizations responsible for keeping the Navy's networks operating and information flowing. Special thanks go to Eric Otte and Britney Chan, who graciously allowed me to conduct multiple iterations of testing at their lab and provided a level of technical expertise I have not found anywhere else. Your efforts to make the fleet a better place for the warfighter are greatly appreciated. I take comfort in knowing that, because of your efforts, the future of Navy networks is a bright one.

Of course, this entire journey would not have been possible if it were not for my advisors, Dr. Don Brutzman and Scot Miller. Dr. Brutzman, thank you for continuing to think big thoughts in order to prepare the next generation of warfighters with the tools they will need to succeed. Scot, thank you for guidance and mentorship. Not only were you instrumental in helping me lateral transfer into the Information Professional community, you have also continued to guide me in the right direction during my time at NPS.

Finally, thank you Bruce Hill for sharing this journey as a fellow student and friend and for helping ensure my sanity remained intact.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

If you are accustomed to living in a world connected by the Internet, your Internet service provider (ISP) may have convinced you that your consumer experience can be maximized by increasing the speed of your connection through the addition of more bandwidth, commonly measured in megabits per second (Mb/s). Tiered pricing plans offered by ISPs such as AT&T, Verizon, and Comcast enable their customers to increase their speed simply by paying more money. Unfortunately, the association of bandwidth as the primary component of network speed is a common misconception. Bandwidth best describes the capacity of a network connection, whereas speed is the calculated rate of motion determined by a distance traveled during a measurable period of time. Travel over the Internet, which is often called the information super highway, is similar to travel over a physical highway used by vehicles. Travel from point A to point B does not occur instantaneously, but, rather, has a certain delay associated with it. The delay, when data travels from point A to point B over the network, is called *latency*. When shipping goods across the country, or delivering data across the network, the rate of delivery is known as *throughput*, and is defined as a measure of quantity over a period of time. Physical shipping may be measured in tons per day, gallons per hour, or cars per minute. Throughput of data transmitted over a network is measured in bits per second. Network performance ultimately becomes an intricate balancing act between capacity, latency, and throughput.

This chapter describes the problems faced by Navy afloat units as they struggle to maximize network performance, even in the presence of increased network capacity. The motivation behind this research is to demonstrate that significant network performance improvements can be achieved by harnessing the capabilities of existing technologies. This chapter will also define research questions and methods to support the hypothesis.

A. PROBLEM STATEMENT

Navy Information Dominance (ID) hinges on three fundamental capabilities: assured command and control (C2), battlespace awareness, and integrated fires; however, none of these are possible without effective communications links (Chief of Naval Operations for Information Dominance, 2013). To support these capabilities, the Navy leverages data generated by ever-more sophisticated sensors, drones, and other network-centric sources that provide situational awareness (Porche, Wilson, Johnson, Tierney, & Saltzman, 2014). Networks, and more specifically, the information flowing through them, are now a center of gravity for the fleet (Chief of Naval Operations for Information Dominance, 2013). In the case of afloat units, this information is carried primarily over satellite communications (SATCOM) paths. To support the demand, the Department of Defense (DOD) continues to increase the capacity of its wideband SATCOM systems. The Wideband Global SATCOM (WGS) system can supply 10 times the capacity of the current Defense Satellite Communications System (DSCS) (Kumar, Taggart, Monzingo, & Goo, 2005).

More capacity alone, however, is not sufficient to overcome reduced wide-area network (WAN) performance due to latency (Belshe, 2010; Grigorik, 2012). Data traveling over SATCOM paths rely on the existing Transmission Control Protocol/Internet Protocol (TCP/IP) based Internet architecture, which provides reliable end-to-end delivery of data (Fall, 2003). Unfortunately, performance of the Internet's Transmission Control Protocol (TCP) can be degraded by high latency found in SATCOM paths, which reduces the overall throughput of the link (Henderson & Katz, 1999). The latency inherent in SATCOM systems reduces the maximum achievable throughput of data for afloat bandwidth allocations below the rates commensurate with equivalent terrestrial bandwidth, causing Navy afloat units to operate under degraded network performance. While reducing latency is harder than increasing bandwidth, reducing latency and bottlenecks over high bandwidth communication paths is essential to improving actual throughput (Belshe, 2010; Grigorik, 2012; Kay, 2009).

B. MOTIVATION AND PURPOSE

To date, a panacea for reducing network latency has not been found, but, rather, improvements rely on the combination of various techniques and technologies. Part of the solution to combating reduced throughput due to high latency is the use of lossless compression (Thompson, 2004). The most common compression tools use variants of the Lempel-Ziv 1977 (LZ77) algorithm to perform compression (Deutsch, 1996; Snyder, McGregor, & Brutzman, 2009). Network-based solutions designed to overcome the contributing factors to latency in WANs can be found in WAN optimization controller (WOC) systems. These systems implement Lempel-Ziv (LZ) based compression as one of many optimization techniques for mitigating latency. A WOC positioned at the local area network (LAN)/WAN boundary is able to apply compression across a broad domain of data, generated by various systems and applications running on the LAN. A second feature of a WOC is the ability to perform data deduplication, a special form of compression, across all transmitted data resulting in the elimination of identical or similar data being transmitted more than once across the WAN. Network-based compression has proven very effective, but it is not the only location where compression can be applied.

Host-based compression implements LZ compression on the data produced by various software applications. LZ compression performed at the application by the host does not significantly outperform similar network-based compression; therefore it is more advantageous to install a single device on the network rather than implement compression at each individual application on numerous hosts. The compaction achieved by LZ-based compression has been surpassed by a new method. Efficient XML Interchange (EXI) is a recent, improved information encoding method that incorporates lossless compression and focuses on the increasingly structured nature of data through the use of the Extensible Markup Language (XML) (W3C, 2014). The rapid adoption of web applications and media-rich Internet applications has made XML a foundational pillar for data interchange (Peintner, Kosch, & Heuer, 2009). EXI is an alternate encoding of XML data, which, in some cases, results in files that are smaller than 10 percent the size of the original XML file (Bournez, 2009).

The purpose of this research is to identify the WAN payload reduction provided by host-based compression and network-based compression when implemented separately, and the changes that occur when implemented together. Current compression implementations have been instrumental in improving network performance; however, there is potential for even greater performance through the combined use of WOCs and EXI.

C. RESEARCH QUESTIONS

Research questions are designed to focus the testing methodology on specific network-based and host-based features. The features tested include EXI (host-based), Riverbed Steelhead's LZ Compression (network-based), and Riverbed Steelhead's Scalable Data Referencing (SDR) (network-based). The following questions are used to guide the methodology and testing conducted in support of the hypothesis:

1. Can EXI provide greater compression than Riverbed Steelhead's LZ Compression for XML data?
2. Does compactness using Riverbed Steelhead's LZ Compression vary with compression level?
3. What effect does Riverbed Steelhead's Adaptive Compression have on Gzip and EXI compressed data?
4. What effect does Riverbed Steelhead's LZ Compression + SDR have on XML data and EXI compressed data when transmitted as a cold transfer?
5. What effect does Riverbed Steelhead's LZ Compression + SDR have on XML data and EXI compressed data when transmitted as a warm transfer?
6. What effect does Riverbed Steelhead's SDR have on XML data and EXI compressed data when a single modification is made at the beginning, middle, and end of a file?
7. What effect does Riverbed Steelhead's LZ Compression + SDR have on XML data and EXI compressed data when a single modification is made at the beginning, middle, and end of a file?
8. What effect does Riverbed Steelhead's SDR have on XML data and EXI compressed data containing more than one modification between files?
9. What effect does Riverbed Steelhead's LZ Compression + SDR have on XML data and EXI compressed data containing more than one modification between files?

D. HYPOTHESIS

Host-based data optimization through EXI, and network-based optimization through Riverbed Steelhead, can be implemented in a complementary manner to achieve greater network performance than either technology is capable of on its own. The combined benefits of both existing technologies provide a path to improved network performance for afloat units without the added cost associated with additional capacity.

E. METHODOLOGY

Practically speaking, the ideal environment for testing is an operational network on board an operational ship using a live SATCOM connection. The ideal data set for testing is real-time data transmitted in an operational environment. Unfortunately, time, money, and other limiting resources precluded this from occurring. Therefore, a simulated network was utilized to transfer a subset of data representative of that found in the operational environment.

Test data formats used were XML, EXI, and Gzip. Test data was transferred over the network to support each of the research questions by selectively enabling the WOC's capabilities for Adaptive Compression, LZ Compression, SDR, and LZ Compression combined with SDR. WAN traffic values were recorded and compared with LAN traffic to arrive at a compression ratio, describing the WAN traffic in terms of times smaller than the LAN traffic. The image in Figure 1 depicts a high-level simplified view of the network configuration and location of optimization techniques.

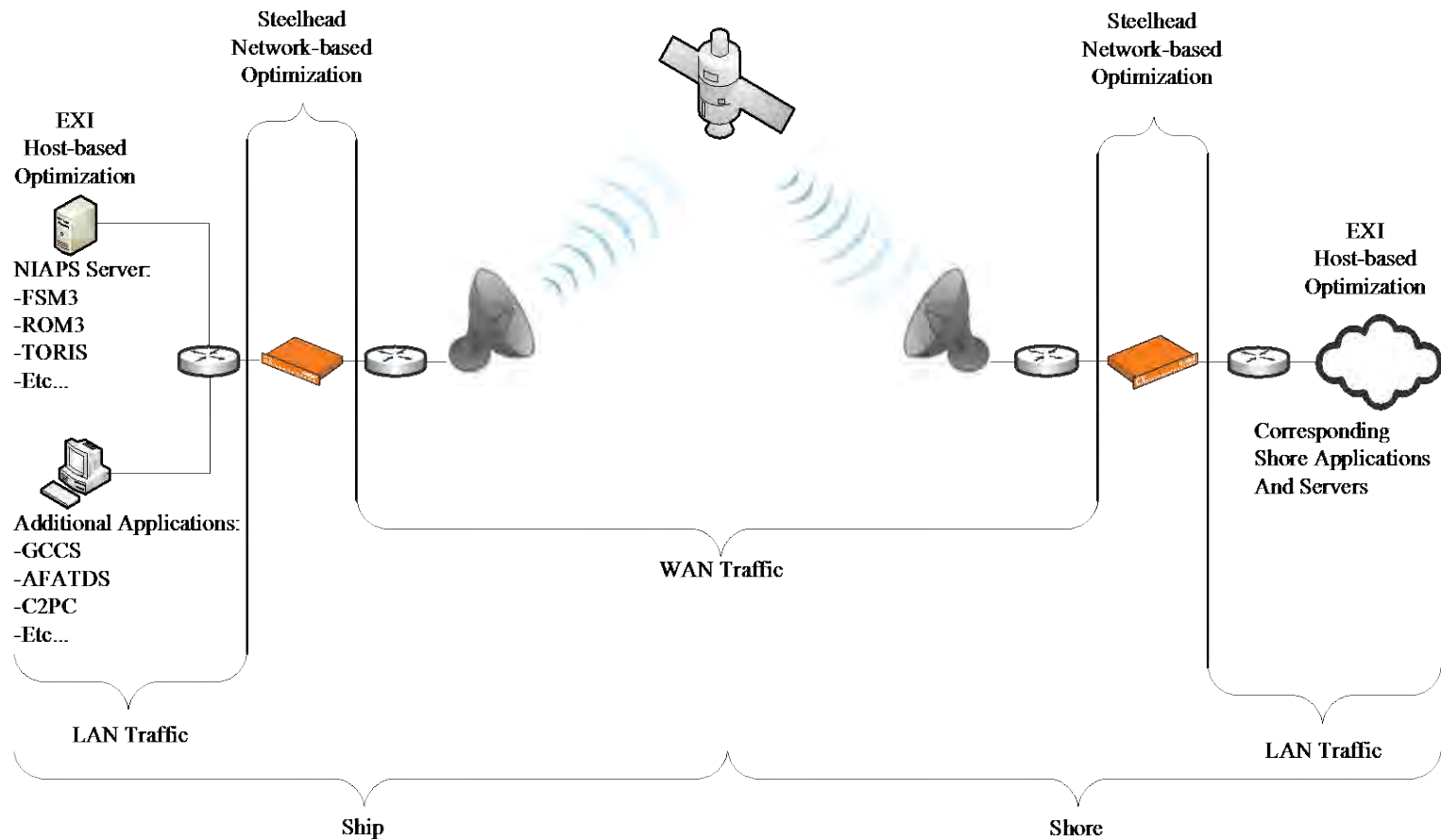


Figure 1. Simplified network diagram depicting ship and shore components of EXI host-based, and Steelhead network-based optimization.

F. BENEFITS OF STUDY

A benefit of this study is to provide a resource for decision-makers that illustrates potential gains achieved by adopting both Riverbed Steelhead WOCs and EXI in a complementary fashion. Fleet testing demonstrated the benefits achieved by using Riverbed Steelhead WOCs, resulting in a decision to procure and install the devices at Network Operations Center (NOCs) and aboard Navy ships. The benefits of EXI have also been demonstrated through DOD testing (Schneider, 2008). EXI implementations are found extensively in the commercial sector and are also found in the Marine Corps, Air Force, Army, and Department of Homeland Security (J.C. Schneider, personal communication, March 15, 2011). Unfortunately adoption of the capability into Navy systems has not progressed beyond that of initial test implementations. The benefits of WOCs and EXI when considered separately each warrant adoption; when combined, there is no denying that the intelligent decision is to use both.

WOCs are limited in their ability to optimize data through compression when the data they receive is encrypted. More applications are encrypting their data before it departs the host, but are no longer providing any type of compression. WOCs such as Riverbed Steelhead can be configured to decrypt, compress, and re-encrypt data to continue providing optimization through compression; however, that capability requires careful consideration of information assurance (IA) requirements and performance desires. The gap in network performance due to the absence of compression prior to encryption can be filled through the use of EXI. This study provides a comparison of benefits achieved

Lastly, this study highlights the need for better network-traffic monitoring tools to build an accurate traffic model of the Navy and DOD's network usage. Only by understanding what applications and data sources are consuming network resources can we attempt to apply the proper optimization and acceleration methods. Presented with examples of potential gains achieved by EXI and Riverbed Steelhead, Navy and DOD leaders can justify allocating the resources necessary to arrive at an auditable bandwidth budget.

G. THESIS ORGANIZATION

Chapter II provides background consisting of foundational network terminology, network obstacles, and details centered on the techniques used to overcome those obstacles. Additional focus is provided on compression, particularly with respect to various implementation methods, some of which result in performance tradeoffs. Chapter III discusses the testing methodology for this research into how EXI can be used to further reduce network traffic and assist with WAN optimization. Chapter IV provides results and analyses of experimentation, identifying the most and least beneficial pairing of host-based and network-based compression. Chapter V presents conclusions and makes recommendations for future work that is needed to incorporate EXI into the Navy and DOD networks.

A portion of Chapter II (specifically, Chapter II, Section G) addressing XML Verbosity and binary XML encodings is a collaborative, co-written product and also appears in *Evaluation of Efficient XML Interchange for Large Datasets and as an Alternative to Binary JSON Encodings* (Hill, 2015). For additional information on EXI performance for large XML files and a compaction comparison of EXI to JavaScript Object Notation (JSON), refer to that document.

II. RELATED WORK

This chapter conducts a review of related work and literature from academic, government, and commercial research to provide a fundamental understanding of the terminology and technology necessary to orient the reader to the network optimization environment. First, a distinction is made between the characteristics of a LAN and those of a WAN. SATCOM is identified as the primary WAN connection utilized by Navy ships afloat, not by choice, but rather by necessity. Next, a detailed discussion of network traffic obstacles, the optimization methods used to overcome them, and the devices called WOCs, which implement them, are explored. The last optimization method, compression, provides benefits that cascade across all other techniques and warrants a more in-depth discussion.

Subsequently compression is further broken down into general compression methods to include data deduplication, and format specific compression methods. Next, while compression has many benefits, it also has limitations. Compression limitations are identified and include, a potential to expand data when applied more than once, little benefit for encrypted data, and security vulnerability when paired with encryption used to establish a secure web browsing connection known as Secure Sockets Layer (SSL), later renamed Transport Layer Security (TLS). The removal of compression from SSL/TLS connections results in a large amount of web traffic that is no longer compressed by the protocol and cannot be compressed by WOCs. The lack of compression over secure connections for the most ubiquitous of web traffic, XML, refocuses this chapter on the last topic, XML compression. XML provides excellent structure for data. However, XML is verbose by design has been compressed using generic and binary encoding approaches. EXI is identified as a successor to generic and binary encoding approaches taking advantage of existing structuring of data to provide superior compression and decompression performance. A brief discussion of its core techniques is provided.

A. LAN VERSUS WAN

Networks can be classified based on many characteristics such as location, size, protocols used, security relationships, or architecture, to name a few. McMillan (2012) identifies that LAN and WAN definitions differ between various texts and offers the definition of a LAN as a high-speed data network covering a single physical location and a WAN as a collection of LANs connected over a WAN technology allowing it to function as one large network. For the purpose of this research, a U.S. Navy ship encompasses the physical location of the LAN. The WAN technology enabling a ship's LAN to function as part of a larger network is achieved through the use SATCOM paths.

Grevers and Christner (2008) identify that network capacity has steadily increased over the last 20 years; however, the cost per bit/second is dramatically lower for LAN than for an equivalent amount of WAN capacity, creating a disparity. The term high-speed is relative when describing LAN connection speeds, but, in most cases, it indicates at least 10 Mbps, and more commonly 100 or 1,000 Mbps (McMillan, 2012). These LAN speeds are representative of those commonly found on Navy ships. Due to the extended distances WAN connections must travel, they are often leased from a telecommunications provider. Long-haul capacity varies depending on the type of connection.

The changes in data transfer rates across a LAN versus a WAN ultimately impact the performance of applications operating across the network. As the network path grows in length, the time required to send data, and receive acknowledgment of receipt, known as round-trip-time, increase. A performance parameter known as the bandwidth delay product (BDP) is calculated by multiplying the capacity of the network path, measured in bits per second, by the round trip time measured in seconds. A network with a bandwidth delay product greater than 100,000 bits is known as a long fat network (LFN), with long referring to the latency and fat referring to the capacity (Jacobson, Braden, & Borman, 1992) Jacobson et al. (1992) identify high-capacity satellite channels as LFNs. When pier-side, Navy ships have the capability of connecting to WAN services provided over copper or possibly fiber optic connections. Once deployed, however, ships have no alternative and must rely primarily on SATCOM paths for their WAN connection, causing them to experience degraded performance associated with an LFN.

B. SATCOM

The Navy has experienced significant growth in the amount of bandwidth available to ships over the last decade. To supply the needs of the fleet, both commercial SATCOM and military SATCOM (MILSATCOM) have been utilized. The WGS system is the highest-capacity military communications system in the United States DOD (Boeing, 2013). Each WGS satellite can support transmission rates of 2.1 Gbps to more than 3.6 Gbps depending on data rates and modulation schemes employed by the mix of ground terminals being serviced (Boeing, n.d.). WGS supplements services provided by DSCS and augments the one-way Global Broadcast Service (GBS) through six on-station satellites with three additional satellites to be launched by fiscal year (FY) 18 (Department of the Air Force, 2012). Unfortunately additional satellite capacity alone is not enough to overcome the performance challenges presented by the LAN/WAN disparity. In order to arrive at a solution to the LAN/WAN disparity issue, a more in-depth discussion of network traffic obstacles is needed.

C. NETWORK TRAFFIC OBSTACLES

In order to understand the methods used to improve WAN performance, it is necessary to understand what obstacles are associated with the delivery of network traffic. Network traffic obstacles are classified into three generalized categories: Network and Transport, Application and Protocol, Operating System and Hardware (Zhang, Ansari, Wu, & Yu, 2012). These categories are not isolated, but rather interrelated where changes in one can result in changes in others.

1. Network and Transport

Network performance is described using a number of different performance parameters. Some are similar to each other and are often misused. Speed is the most generic performance term used in networking and its meaning is highly dependent on the context in which it is used. Rated speed is often listed as the number of bits per second and is the biggest magic number in networking. Using a nominal speed rating to describe a network or a device on the network only theoretically describes the network and is an incomplete description. Other real-world factors influence the performance of the

network (Kozierok, 2005). Those factors include bandwidth and channel capacity, latency, congestion and packet loss, and throughput.

a) Bandwidth and Channel Capacity

A common mistake is to use the performance factor bandwidth to describe the speed of data transmission. From the scientific study of electromagnetic radiation, bandwidth refers to the width of a band of frequencies used to carry data. From the network perspective, bandwidth is a widely-used term for the data-carrying capacity of a network or a data-transmission medium (Kozierok, 2005). The channel capacity, as described by Claude Shannon in units bits per second is a function of both bandwidth (in hertz) and the ratio of signal to noise (measured in watts) over the transmission medium (Couch, 2013). Channel capacity presents the upper transmission limit that may be achieved while keeping the probability of errors near zero (Couch, 2013). Transmitting data over the channel is achieved through various modulation and encoding schemes. For instance, a system using 16-phase shift keying (16PSK) will achieve a higher transfer rate measured in bits per second than a system using binary-phase shift keying (BPSK). The first system, however, will require a higher signal-to-noise ratio than the second. Typical SATCOM allocations assigned to navy ships include a small power margin used to overcome attenuation caused by clouds and mild weather occurrences. The SATCOM modem used by Navy ships is configurable to exploit the additional power margin and achieve higher bandwidths through the use of more aggressive modulation schemes, however, the capability is not currently utilized by the Navy. Pending additional testing by the Space and Naval Warfare Systems Command (SPAWAR), the capability may be used in the future to maximize SATCOM connection capacity.

Bits per second are commonly used for listing the rated speed for network devices. Switches, routers, and modems, among other devices, do not run at their full-rated speeds, but rather run substantially below it due to real-world performance factors (Kozierok, 2005). Introducing real-world performance factors results in the inability to use all of the available system bandwidth, thereby reducing the system to an effective bandwidth. Even if more bandwidth is made available, the effective bandwidth will not differ greatly (Belshe, 2010). Effective bandwidth is derived from the perspective of an

individual user's application or service. Adding more bandwidth may allow for more users and applications to access the communication channel; however, each application's performance is still limited by the effective bandwidth. One of the determining factors of effective bandwidth and often the most detrimental factor is latency.

b) Latency

When people travel from one location to another, they inevitably encounter various obstacles, which delay them in their travels. This delay when discussing network traffic is called latency. Latency is the amount of delay involved in moving information from node to node across the network (Grevers & Christner, 2008). The relative location of the nodes must be considered when discussing latency as a performance factor. Every network connection spans some distance. It may be inches or feet between two network interface cards (NICs) on a router and switch or it may be thousands of miles over a satellite connection. No matter the distance traveled, the time it takes to complete the trip is called propagation delay. The simplest connection includes only two nodes. However, as the network grows, the connections traverse many intermediary connections, all of which add their own delays.

Propagation delay is directly related to the physical separation and propagation velocity. Navy networks rely heavily on SATCOM links to support ships at sea. This results in a large propagation delay of roughly 400 ms in one direction. As the distance between the ship and the satellite ground station increases, so does the propagation delay. Propagation delay only accounts for latency associated with time spent on the network transmission medium. Other important factors contributing to latency are serialization delays, processing delays, and queuing delays (Grevers & Christner, 2008; Grigorik, 2013).

Serialization, also known as transmission delay, is the time required transferring data from a queue to the transmission link. Transmission delay is a function of the data size and the transfer rate of the link. Processing delay is the time required for processing the data, checking for errors, and determining the data's destination. Queuing delay is the amount of time when data is not being either processed or transmitted prior to reaching its final destination (Grigorik, 2013). Each device encountered across the network

contributes to the total latency. Perceived network latency is the sum of all delays as experienced by the application in use and the end user.

c) Throughput

Network capacity and latency together impact the third most common measure of network performance: throughput (Grevers & Christner, 2008). Throughput is the net effective data transfer rate measured in bits per second, and thus can never exceed the available capacity of the smallest segment of the network. Throughput and bandwidth, when describing network capacity in bits per second, are often used interchangeably, even though they are not the same (Kozierok, 2005). Data packet loss reduces effective throughput over the network. Congestion combined with other network events can result in packet loss. Congestion occurs when the rate of data arriving at a network device is greater than the rate of data departing the device. Arriving data is placed in a queue prior to being processed for retransmission. Once the queue is full additional data can no longer be received and, depending on the protocol used, may be lost. This results in data being dropped and potentially retransmitted. Data cannot just be thrown on the network for transmission (Kozierok, 2005). Overhead is added to the data in support of the communication protocols used for transmission. The overhead required by the protocols consumes a portion of the communication channels capacity and in some cases adds to the latency, further reducing the effective throughput.

2. Application and Protocol

Application performance is impacted by limitations associated with the protocols used to transfer data across the WAN (Zhang et al., 2012). The Internet and most common networks today rely on the Internet protocol suite TCP/IP. TCP is responsible for the connection, management, and reliable data transport. The Internet Protocol (IP) is responsible for host-to-host routing and addressing. This family of protocols provides the delivery mechanism for most of today's network traffic and supports Application layer protocols in the TCP/IP protocol stack. Protocols designed for a LAN environment perform poorly when introduced to a WAN environment with reduced capacity, high latency, and reduced throughput (Grevers & Christner, 2008). To achieve reliable data

transport, TCP implements a three-way handshake to establish a connection. The handshake must be complete before data even starts to flow. Conducting the handshake causes TCP to fall victim to twice the amount of one-way latency, known as round-trip time (RTT). Furthermore, TCP must implement slow-start and congestion control mechanisms to sense the channel capacity and recognize when the capacity has been exceeded, resulting in packet loss (Grigorik, 2013). TCPs three-way handshake, slow-start, and congestion control coupled with higher latency over WAN connections result in inefficiencies for applications relying on TCP (Zhang et al., 2012). It is possible to tune the performance of TCP, using a variety of operating system specific techniques; however, tuning requires a high level of expertise (Tierney, 2008).

As stated by Grigorik (2013), while bandwidth continues to increase, latency is bounded by the speed of light causing in most cases, latency, not bandwidth to become the bottleneck in TCP based networks. The User Datagram Protocol (UDP) is an alternate data transportation protocol void of any handshake and congestion sensing or control mechanisms. UDP, however, does not provide any delivery guarantees and is susceptible to one-way latency. Its tolerance to data loss makes UDP the more common protocol choice for streaming audio and video applications.

3. Operating System and Hardware

The selection and pairing of operating system and hardware at the client and server can impact an applications performance over the network. Various operating systems implement the TCP/IP protocol suite with their own set of optimizations. Hardware and software configuration of a server and client constitute the first source of latency as application data moves down the protocol stack and through the hardware necessary to initially place it on the transmission medium. Latency from these sources can include hard drive and memory access times as well as the speed and number of processes on board.

D. WAN OPTIMIZATION TECHNIQUES

As defined by Gervers & Christner (2008), WAN optimization is a set of services that overcomes the performance limitations caused by transport protocols, network conditions, and network utilization. The most common techniques used to provide the foundation for WAN optimization are protocol optimization, caching, prefetching, data deduplication, and compression (Deng & Manoharan, 2013; Grevers & Christner, 2008; Zhang et al., 2012). Other techniques exist and new techniques continue to emerge, however they are commonly a subset of those listed or are niche technologies with narrow application.

1. Protocol Optimization

Protocol optimization is the use of in-depth protocol knowledge to improve inefficient protocols by making them more tolerant to high latency in WAN environments (Zhang et al., 2012, p. 1097). Common protocols requiring optimization for the WAN environment are TCP, Hypertext Transfer Protocol (HTTP), Common Internet File System (CIFS), Messaging Application Programming Interface (MAPI), and SSL/TLS. Protocols such as TCP, HTTP, and SSL/TLS experience low efficiency over a WAN, simply because they were originally designed for a LAN environment; other protocols such as CIFS and MAPI are chatty in nature, requiring extensive messaging before useful application data is ever transmitted (Grevers & Christner, 2008; Zhang et al., 2012). Protocol optimization can be implemented through a WOC or it may occur over time as the protocol evolves and matures as is seen with HTTP. The two current versions of HTTP are 1.0 and 1.1 with HTTP 2.0 in draft format (Belshe, Thomson, & Peon, 2014; Zhang et al., 2012). SPDY, pronounced “SPeeDY,” is an experimental protocol initiated by The Chromium Project at Google geared toward improving HTTP; concepts from SPDY are implemented in the HTTP 2.0 draft (Belshe et al., 2014; The Chromium Projects, 2014).

2. Caching

Caching reduces bandwidth consumed by placing an intermediary buffer between the slower WAN connection and the client or server, thereby reducing WAN traffic and user-perceived latency (Grevers & Christner, 2008; Zhang et al., 2012). In this author's experience, caching performance is impacted by capacity, location relative to the authoritative data source, and the content freshness. Content freshness is the amount of time that the cached data accurately reflects the source data. Content validation prevents stale content from being served and is a required function for a WOC to perform caching (Grevers & Christner, 2008). Caching only provides benefit for data that is requested more than once; otherwise the data consumes storage capacity that may otherwise be used for data in higher demand (Zhang et al., 2012, p. 1095). Caching implementations vary depending on the performance needs and configuration of the network. Zhang et al. (2012) divides caching into three broad categories: location of cache, cooperation, and type of cached object. Each category is further broken down based on additional configuration characteristics. Caching as described here is considered passive and has been found to reduce latency by 26% at best (Zhang et al., 2012). The specifics pertaining to the variety of caching methods and configurations is beyond the scope of this research. To improve performance beyond passive caching, proactive caching can be used, sometimes referred to as prefetching (Zhang et al., 2012).

3. Prefetching

Prefetching increases data exchange rates by proactively retrieving data in anticipation of use based on history or content predictions (Zhang et al., 2012). User-perceived latency is reduced by using otherwise idle time to download data before it is requested (Deng & Manoharan, 2013). Zhang et al. (2012) identify that prefetching derives its performance based on the location of the prediction and prefetching engines, which utilize algorithms classified as history-based prediction and content-based prediction. Additional details pertaining to prefetching and prediction engine operation and implementation are beyond the scope of this research.

4. Compression and Data Deduplication

Compression and data deduplication greatly reduce the amount of traffic transferred over the network. Deng and Manoharan (2013, p. 22) describe data deduplication as a specialized form of compression while Zhang et al. (2012, pp. 1092–1093), express each as separate techniques. Data deduplication, perhaps better defined by Grevers & Christner (2008) as data suppression, is a function that utilizes a shared codebook between sender and receiver to eliminate the transfer of redundant data across the network, whereas compression implements algorithms to analyze and consolidate a predefined group of data known as a window. In a recent report released by the Center for Naval Analysis (CNA), Bentrup, Otte, Chan, Vavrichek and Gingras (2012, pp. 13–14), describe compression and data deduplication as “algorithm-based” and “disk-based” respectively. Compression and data deduplication are implemented in a complementary fashion. The first transfer of data is not seen as redundant but can be reduced in size using compression. Additionally, the shared codebook symbols and the data they represent may be compressed, resulting in additional reductions in local storage as well as total data transferred (Grevers & Christner, 2008, Chapter Three). Additional details on compression are covered in Chapter II, under heading G.

E. WAN OPTIMIZATION SOLUTIONS

Many organizations have transitioned from executing applications, and accessing data over the LAN, to methods incorporating Software as a Service (SaaS) and other cloud-based computing solutions, causing demands on WAN connectivity (Nirmala, 2014, p. 282). The divide between LAN and WAN network performance has resulted in a growing demand for WOCs, which implement one or more acceleration techniques to overcome the bandwidth disparity and reduce latency (Grevers & Christner, 2008). The WAN optimization market has grown from \$1 billion in 2008 to \$4.4 billion in 2014 (Nirmala, 2014). WOCs are commonly deployed symmetrically or asymmetrically as dedicated hardware appliances, virtual appliances, or as software clients running on individual clients (Skorupa & Munch, 2014). Gartner analysts Skorupa & Munch (2014) evaluated 13 WOCs using 15 evaluation criteria, ultimately identifying Riverbed and

Silver Peak as industry leaders, and Cisco as an industry challenger. Additionally, SPAWAR, supported by CNA, as part of an Office of the Secretary of Defense (OSD) Defense Acquisition Challenge, tested five systems in a laboratory, which models a realistic afloat communications environment; Riverbed, Silver Peak, and Cisco emerged as the most promising WAN optimization devices for afloat networks (Bentrup et al., 2012, pp. 1–2).

For nearly ten years, the Navy has implemented the WOC PacketShaper on ships and at the NOCs to provide quality of service (QoS) functions and traffic compression. PacketShaper, developed by Blue Coat Systems, is identified as a “niche player” by the Gartner report for reporting, traffic control, and compression (Skorupa & Munch, 2014). Testing conducted by Bentrup et al. (2012), has shown Riverbed Steelhead’s disk-based compression to surpass the memory-based compression capabilities of PacketShaper, resulting in the planned incorporation of Riverbed appliances into the Automated Digital Network System (ADNS) INC III starting with service pack three. Bentrup et al. (2012), and Nirmala (2014) identify better compression methods as instrumental to improve WAN optimization. Improvements achieved through compression inevitably impact protocol optimization, caching, prefetching, and data deduplication by reducing the amount of bits that each technique must process.

F. COMPRESSION

The two principle types of compression algorithms are lossless, meaning the original message can be reconstructed exactly from the compressed format, and lossy, meaning that the decompressed or restored data may only be an approximation of the original message (Blelloch, 2013; Thompson, 2004). Lossy compression is better suited for data such as audio and video associated with teleconferencing where a loss in resolution may be undetectable or at least acceptable (Blelloch, 2013). Lossless compression is utilized for text and other business data where even the change in one bit is unacceptable (Thompson, 2004). Deng and Manoharan (2013) further divide compression into the subcategories of general compression and format specific compression.

1. General Compression and Data Deduplication

The general compression algorithm, Lempel-Ziv (LZ), or one of its variants, are the most common algorithms implemented in WOCs including Riverbed Steelhead (Grevers & Christner, 2008; Riverbed Technology, 2013b). LZ-based algorithms build a dictionary of previously seen strings by using a sliding window to scan through an object (Blelloch, 2013; Grevers & Christner, 2008). LZ compression combined with Huffman coding is used to form composite compression algorithms such as DEFLATE, which is implemented in many common applications such as Gzip, HTTP, and Adobe's Portable Document Format (PDF) format (Kattan, 2010).

Grevers and Christner (2008) define the compression domain as the limit of an algorithm's effectiveness dictated by the capacity of the sliding window, granularity of identified patterns, and the structures that map between signatures and previously seen data. Without any modifications, the encoder forgets a compressed object once it is transmitted even if the same object appears twice in a row (Bentrup et al., 2012, p. 13). Data deduplication leverages a much larger static compression history by expanding the compression domain beyond the packet, file, or session being transferred. Data deduplication, implemented at the network or transport layer, does not discriminate between applications; therefore, duplications can be removed between an object downloaded via a website and the same, or modified object transmitted via email as an attachment (Grevers & Christner, 2008).

Riverbed's proprietary algorithm for implementing data deduplication is called Scalable Data Referencing (SDR) (Riverbed Technology, 2013a, p. 16). Bentrup et al. (2012, p. 14) describe the process for referencing data segments similar to the one used by SDR in Figure 2.

[Figure 2] is an example of referencing data segments and building them into larger segments. In this example, a file (or portion of a file) is initially split into nine segments. As the system learns the data patterns, it combines segments 1 and 2 to form segment 10, and segments 3 and 4 are combined into segment 11. With additional pattern learning, segments 10 and 11 are combined into segment 100. Using this approach, the entire file (or portion of a file) can be passed with three data references: 100, 6, and 12. Later if segment 8 changes (such as a slide in a PowerPoint file that is

edited and saved), the entire set of data can be passed by references 100, 6, 7, a dynamically compressed version of the new segment 8, and reference 9. These references and segments are stored on a disk in a type of data dictionary that can be used to retrieve matches in the future, along with the reference number to use.

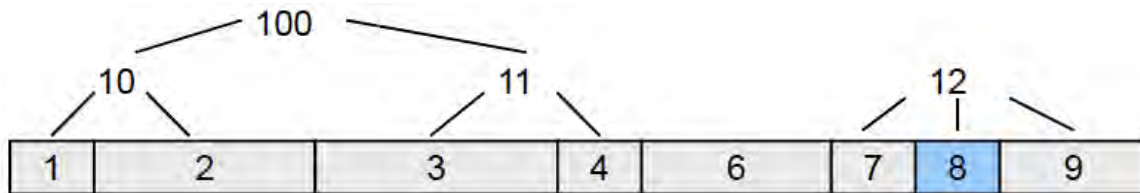


Figure 2. Passing disk-based compression data segments via reference number (from Bentrup et al., 2012, p. 14).

The Steelhead Appliance Deployment Guide (2008, p. 16) states that the Steelhead appliance continuously builds the data store to include more and more data references as files are copied, edited, renamed, and otherwise changed or moved.

2. Format Specific Compression

Deng and Manoharan (2013, p. 22) indicate that while there is no best-fit solution for all situations, format-specific compression achieves higher compression ratios than general compression for some file formats. Specific formats commonly requiring compression are audio, video, still graphics and text. General compression used in WOCs is applied primarily at the Network and Transportation layer; however, format specific compression is applied at the Application layer. Deng and Manoharan (2013, pp. 22–23) draw particular attention to XML as it poses significant overhead that when compressed can contribute to WAN optimization. The graphic in Figure 3 is provided to orient readers, not familiar with the Open Systems Interconnection (OSI) and TCP/IP networking models, to the different network abstraction layers where compression may be applied.

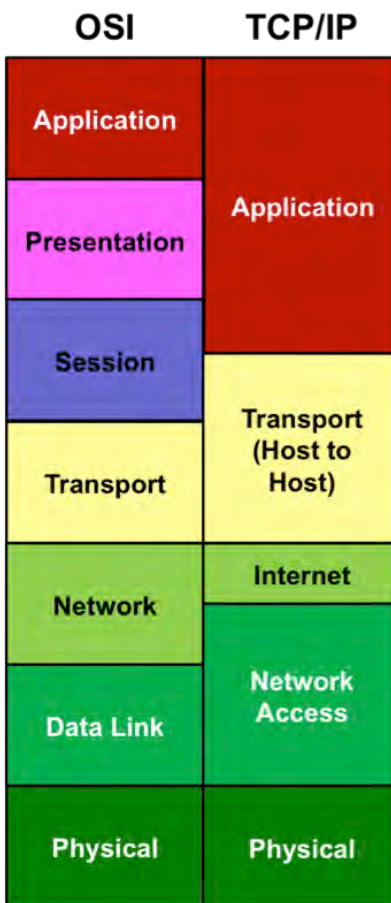


Figure 3. Comparison between the OSI and TCP/IP networking models. Host-based format specific compression is generally applied at the application layer. Network-based general compression is commonly applied at the Network and Transport layers. (after J. Green, personal communication, 12 December 2014).

3. Compression Limitations

Compression offers significant performance improvements for WAN traffic but is predominately subject to two conditions where the effects of compression are reduced or negated. The first condition is the application of compression to already compressed data. The attempt to compress data, which has previously been compressed, may result in an increase in size rather than further reduction in size (Morgan & Dennis, 2003). It is desirable for compression implementations to identify when compression has previously been applied to data and forego the application of any additional compression, saving processing time, and eliminating any unnecessary increase in size.

The second and more detrimental condition reducing the potential benefits of compression is the application of encryption to data before the application of compression. Order of operations has significant impact on the effectiveness of compression, which must occur before encryption is applied (Morgan & Dennis, 2003). A detailed understanding of the data's transmission path is required in order to identify where encryption is applied and where compression is applied. The need for WOCs to accelerate a growing amount of encrypted traffic is not a new challenge, as identified by Betts in his article, *SSL Traffic Clogs WANs* published in 2007 (Betts, 2007). The release of classified documents by Edward Snowden created a catalyst for more Internet companies and users to implement data encryption (Curran, 2014). Adoption of encryption through SSL/TLS is forecasted to increase causing higher demand for devices capable of providing efficient delivery of those traffic flows (Intel, 2013).

Early implementations of SSL/TLS, citing the verbose size of XML and its growing use on the Internet as motivation, included options for compression using the DEFLATE specification (Hollenbeck, 2004). Kelsey (2002) describes an information leak created through a side-channel when pairing compression and encryption, which reveals "information about their inputs by the size of their outputs." Meyer and Schwenk (2013) further explain that the Compression Ratio Info-Leak Made Easy (CRIME) attack tool exploits the side-channel vulnerability enabling a skilled attacker to decrypt traffic, enabling cookie stealing and session take-over. Due to the security vulnerability exploited by CRIME, browsers have removed the option for compression over TLS connections (Clark & Van Oorschot, 2013, p. 513) .

The removal of a native compression capability for SSL/TLS connections results in encrypted data flowing over network paths, which has not been compressed, and cannot be compressed without first decrypting the data. Decrypting the data at any point other than the intended receiver presents potential IA vulnerabilities that must be addressed. If a WOC cannot "see" into the data passing through it, due to encryption, then optimization through compression cannot be achieved. WOCs such as Riverbed Steelhead have since introduced capabilities to decrypt, compress, and re-encrypt data in order to achieve optimization through compression. The decision to enable optimization

of encrypted data requires a careful analysis and comparison of performance improvements and IA vulnerabilities. It is important to understand the distinction that SSL/TLS traffic can still be accelerated through protocol optimization, however the lack of compression is the limiting factor in achieving performance equal to that of unencrypted traffic.

G. XML COMPRESSION

For networked systems with limited throughput, memory or battery power, XML has an Achilles heel; it is not, and was never designed to be, a compact encoding. This section summarizes the design decisions leading to XML's verbosity, and previous work addressing the issue. As described in Chapter I, subsection G, this section was co-written with Bruce Hill and therefore also appears in his thesis (Hill, 2015).

1. Verbose by Design

XML's designers aimed largely to streamline data transfer on the web and to eliminate the complexities of its predecessor, Standard Generalized Markup Language (SGML) (Kangasharju, 2008, pp. 13–14). They built XML to be simple for humans to use, write, and read, implying that it must be a plaintext format; reading and debugging binary documents is near impossible without computer assistance (Bos, 2001; Bray, Paoli, & Sperberg-McQueen, 1998). The specification met its goals, and the plaintext encoding of XML is part of the reason behind its success.

A drawback to the simplicity of plaintext encoding is an associated size increase. The original XML specification states that “terseness in XML markup is of minimal importance” (Bray et al., 1998). In practice, however, computers are often the only entities processing XML messages, so for many applications, terseness is more desirable than human-readability. To clarify the issue, consider the word “Efficient.” Encoded in plaintext using UTF-8, each of its nine letters uses 1 byte, or 8 bits, for a total of 72 bits. In a binary encoding, those 72 bits can represent $4,722,366,482,869,645,213,696(2^{72})$ different words, which is approximately 7.9 trillion times as many words as there are in the English language (Oxford University Press, 2013). Considering this overhead, there certainly are more efficient ways to communicate the word “Efficient.”

2. Generic Compression Approaches

A common approach to reducing XML transfer sizes is to apply a generic, lossless compression algorithm for which the sender and receiver use an agreed upon coding/decoding program, commonly referred to as a codec. Nearly all operating systems include software implementations of the DEFLATE algorithm and can process file formats such as Zip and Gzip, so those are common in practice. In general, Gzip compression offers significant compaction over plaintext-encoded XML, decreasing it to 50% or less of original size, though the compaction rate varies based on the XML contents (Bournez, 2009). In a few cases, however, Gzip encodings increase the file size (Bournez, 2009).

When clients on the web send HTTP requests to servers, they may specify that they accept responses in specific encodings (Fielding & Reschke, 2014, p. 40). If the server supports that encoding, it applies the requested compression prior to sending its response. The Internet Assigned Numbers Authority (IANA) maintains an official list of formats, though clients may choose to request others (Internet Assigned Numbers Authority, 2014). Both the Apache HTTP server and the Microsoft Internet Information Services (IIS) server support Gzip compression without extensions (Microsoft, 2015; The Apache Software Foundation, 2015). If the server does not store compressed copies of the requested resources, it must spend time applying compression before responding, which presents a series of tradeoffs based on network speed, server capabilities, and traffic load (Morse, 2005).

3. Binary Encoding Approaches

Binary encodings are another group of techniques for compacting XML data. Generic compression algorithms such as Gzip make no assumptions about the data to which compression will be applied; they work on any stream of bytes. XML documents, however, have a well-defined tree structure. Binary encoding algorithms designed specifically for XML documents use *a priori* knowledge of this structure to achieve greater compaction than generic algorithms (Sakr, 2009). Between the publishing of the XML recommendation in 1998 and 2014, multiple standards and formats emerged, each

specifying such a binary representation of XML. The following paragraphs briefly describe select methods illustrating techniques relevant to this research. Sakr (2009) provides a broader survey of formats along with comparative test results.

One category of XML compression methods works by substituting short binary tokens for longer plaintext elements in a process called tokenization. The Wireless Application Protocol (WAP), developed by the WAP Forum, was an early such solution in this group. WAP is a group of standards that define a complete architecture optimized for low-memory mobile devices connected by low-capacity wireless networks (The WAP Forum, 2000). A WAP system routes client requests through a gateway device between a web server and end client. The WAP gateway transforms the requested web page or information into a Wireless Markup Language (WML) document, an XML format. In a process called tokenization, the gateway converts long plaintext elements from the WML document into short, binary tokens that consume less space on the final wireless hop—a format called WAP Binary XML (WBXML) (Martin & Jano, 1999). The image in Figure 4 depicts a common WAP architecture. The Fast Infoset (FI) specification takes a similar approach to WBXML. It uses binary tokens to encode XML structural elements, and builds dynamic vocabulary tables that hold recurring strings of characters (International Organization for Standardization, 2007).

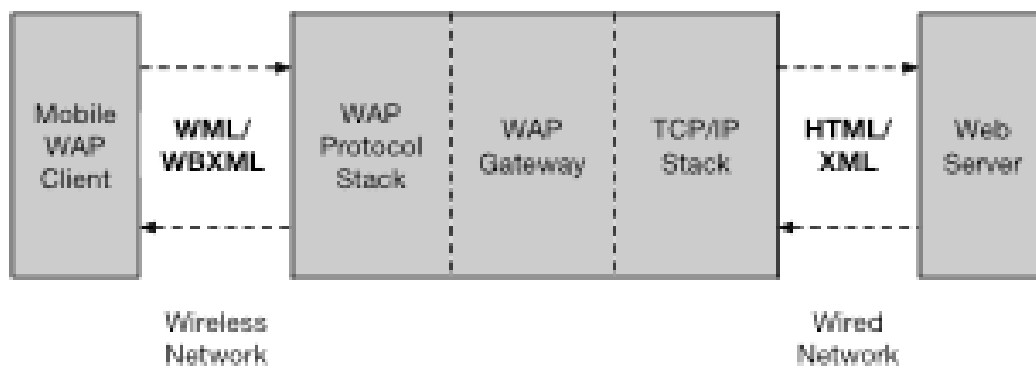


Figure 4. WAP system diagram showing XML transformation into WML via a gateway architecture (after Saha, Jamtgaard, & Villasenor, 2001).

Another common approach to XML compression is to use knowledge of both the XML structure and the mechanics of one or more generic compression algorithms to pre-process the XML document so that conventional compression algorithms such as Gzip perform better as a final processing step (Sakr, 2009). An early example of this method is the XMill compression method, developed by Liefke and Suciu (2000). It separates data from structure, reorganizes the XML document to group similar items together, and finally applies customizable semantic compression if the specific contents of an element are known (Liefke & Suciu, 2000). A possible limitation of this approach is poor compaction for small files—in Liefke and Suciu’s (2000)’s work, files below approximately 20 kilobytes suffered from pre-processing overhead and were better compressed with only Gzip.

4. EXI

Between 2000 and 2014, a wide variety of XML-specific compression techniques were developed; different methods optimized for memory footprint, difficulty of implementation, encoding speed, decoding speed, random access and compactness (Kangasharju, 2008; Sakr, 2009). In light of this complexity, two W3C working groups evaluated a variety of binary XML encodings for adoption as the consortium’s recommended standard. The XML Binary Characterization (XBC) Working Group enumerated a list of target use-cases and an associated list of more than 25 requisite properties for a binary XML standard (Cokus & Pericas-Geertsens, 2005). Table 1 lists the minimum requirements for a binary XML format as determined by the XBC Working Group. Beginning in 2005, the successor Efficient XML Interchange Working Group compared a variety of candidate compression algorithms against the properties, ultimately settling on AgileDelta’s Efficient XML (Le Hegaret, 2005; Schneider, Kamiya, Peintner, & Kyusakov, 2014). In March 2011, the EXI format reached official W3C recommendations status (Schneider et al., 2014).

Table 1. Minimum requirements for a binary XML format as determined by XBC Working Group (after Goldman & Lenkov, 2005).

MUST Support	MUST NOT Prevent
Directly Readable and Writable	Processing Efficiency
Transport Independence	Small Footprint
Compactness	Widespread Adoption
Human Language Neutral	Space Efficiency
Platform Neutrality	Implementation Cost
Integratable into XML Stack	Forward Compatibility
Royalty Free	
Fragmentable	
Streamable	
Roundtrip Support	
Generality	
Schema Extensions and Deviations	
Format Version Identifier	
Content Type Management	
Self Contained	

EXI capitalizes on multiple techniques to improve compression, and they vary depending on a set of EXI options passed to the codec. This section briefly discusses core compression techniques available in the EXI specification. Hill (2015) covers additional option configurations in his methods chapter. Kyusakov (2014) and Peintner and Pericas-Geertsen (2007) offer concise explanations of the EXI algorithm, and the official EXI Specification presents a normative description (Schneider et al., 2014).

a) Grammar-Based Encoding

XML documents have a set of rules governing their structure. While processing an XML document, an XML parser interprets the various tags as events, which can be considered as state transitions in a grammar. At any given point in a document, an XML parser can expect the next event to be one of a finite set of possibilities with different probabilities. An EXI encoder builds an internal model of the document as a set of such grammars, and assigns short variable length numeric codes to each event (Schneider et al., 2014). The short codes replace longer tag structures in the encoded stream.

b) String Table

While parsing an XML document, an EXI encoder builds a table of character strings, which can be thought of as words. The first time the encoder encounters a word, it writes it in full and assigns it a short numeric code and every time afterwards, the encoder replaces the full word with the shorter code. This technique addresses the repetition of opening and closing tags in XML, as well as documents where the data itself is repetitive.

c) Data Types

If an XML document has a corresponding XML Schema describing the data type of its elements, an EXI encoder can use that information to encode data in a compact format. Without a schema, the EXI encoder treats all element contents as plaintext character strings. For example, consider an XML attribute with a value of “false”. As a plaintext string, its five characters each fill 8 bits, for a total of 40 bits. However, with a schema identifying the attributes as Boolean, the EXI encoder knows that the attribute can only take one of two logical alternative values: “true” or “false”. Since 1 bit can encode 2 different values, the encoder uses that information to shorten “false” from 40 bits to 1 (Schneider et al., 2014).

d) Range Restrictions

In addition to specifying an element’s data type, an XML Schema can define a set of restrictions on its possible values. For example, consider an XML attribute defined as an integer, with a value of 15,009. An EXI encoder by default encodes it with 2 bytes, or 16 bits. However, if the schema restricts the attributes values to the set of integers between 15,000 and 16,000, the EXI encoder simply writes its offset from 15,000 (i.e. the value 9), which uses only 4 bits.

e) Channelization

An EXI encoder can reorganize the contents of an XML document such that similar elements are close together in the output stream, a process called *channelization*. Since many compression algorithms identify recurring strings of characters, and perform better when the recurrences are localized rather than scattered throughout a document, the reorganization optimizes the document for later compression (Salomon, 2008; Schneider

et al., 2014). By default, EXI applies the DEFLATE algorithm for the final compression step, although others can be used.

H. CHAPTER SUMMARY

Demands for higher performance over SATCOM systems force the Navy to increase capacity, reduce reliance, or optimize available capacity in order to meet operational demands. Provisioning additional capacity is expensive and due to high latency associated with orbital distances and propagation time, capacity alone does not guarantee an improvement in performance. Reducing reliance on SATCOM paths is not realistic because more sensors and weapon systems are network centric and require high levels of connectivity. Optimization of existing capacity offers the greatest return on investment.

WAN optimization is achieved through protocol tuning, intelligently positioning data, data de-duplication, and data compression. WOCs enable significant improvements in WAN performance by utilizing general compression algorithms and data deduplication across a broad domain of data, but are still not a panacea for optimizing SATCOM performance. Additional compression methods may provide even further benefits. The advent of a new compression method, EXI, was designed to greatly reduce the size of XML documents, the most ubiquitous format for structured text transferred over the Internet. EXI can further reduce network traffic over SATCOM paths, thereby freeing up existing capacity for use by other less compressible traffic. Where WOC performance may be reduced by the application of encryption prior to data arriving at the WOC, EXI retains its optimization benefits for encrypted traffic by compressing data prior to transmission thereby avoiding the IA tradeoffs associated with encryption implementation over WOCs.

III. METHODS

The ideal environment for testing is the real world using an actively deployed ship's network over a live SATCOM connection. The ideal data set for testing is live data used in an operational environment. Unfortunately, time, money, and other limiting resources did not allow for the ideal environment and data set. Therefore, a high-fidelity simulated network was utilized with a subset of data representative of that found in the operational environment.

A. DATA SET SELECTION

The goal of data selection is to obtain a sample of data, which is an exact copy used in the operational environment. Frequently operational data can be classified, contain personally identifiable information (PII), or have restrictions pertaining to its releasability for other reasons. If an exact copy is not available then a sanitized copy where any non-releasable information has been removed may be used. Since compression performance is partly influenced by the amount of repetition in a data set, the act of sanitation may inadvertently influence overall performance. As a theoretical example, if a data set originally contained social security numbers, which are unique and non-repeating, and the sanitation process replaced them with xxx-xx-xxx, then this data would be identified as a repeating value in the test sample resulting in greater compression occurring. Similar to sanitation, the method used to ensure that test data is of varying size may influence the compression performance. If a data set is simply duplicated multiple times within the original file in order to increase its size, this will result in less than realistic compression performance. Ultimately, testing conducted in anything other than a live operational environment is subject to certain artificiality factors, which may not be possible to completely remove. In order to ensure unrestricted repeatability, no actual PII or sensitive data was utilized in these testing efforts.

Data selection was limited to unclassified and releasable information. Selected data sets are structurally representative of administrative data, and operational target and track data. All data were converted to the EXI format using AgileDelta's implementation

of the EXI standard called Efficient XML (EFX). AgileDelta also provided optimized schemas. To achieve the best compression through EXI, the XML schema may need to be refined beyond what is utilized to only perform validation. The value of a refined schema for compression is addressed by Hill (2015). Three different data sets were used and are described below.

a) Food Service Management—Administrative

Food Service Management (FSM) is an administrative application and is one of many maintenance, logistics, administrative, training, and management applications hosted on the Navy Information Application Product Suite (NIAPS) server aboard navy ships. Test data consisted of an FSM Structured Query Language (SQL) database exported in XML format. The entire export consisted of 94 individual XML files of various lengths. Of the 94 files, four files were selected for testing, with the first file being the smallest, the last file the largest, and the remaining two files falling in between. The Navy Supply Systems Command (NAVSUP) provided the FSM data set. Information for the NAVSUP FSM point of contact is provided in Appendix A.

b) Joint Rapid Architecture Experiment 2006—Operational

The 2006 Joint Rapid Architecture Experiment (JRAE 06) was conducted at SPAWAR San Diego in 2006 and focused, in part, on improving the tactical flow of information to and from the Joint Target Management (JTM) service-oriented architecture (SOA). Participants in the experiment included AgileDelta who through the implementation of their product EFX demonstrated the improved throughput achieved through encoding XML using EXI. Applications used in testing include Advanced Field Artillery Tactical Data System (AFATDS), Command and Control Personal Computer (C2PC), and Global Command and Control System-Maritime (GCCS-M), among others. Data used in this research are a subset of the data utilized during JRAE 06. The data set contained 911 different targets produced by AFATDS.

For this research, the single file containing 911 targets was processed to create four different sets of target batches. The first set contained only one target per file, the second set contained ten targets per file, the third set contained 50 targets per file, and the

fourth set contained 100 targets per file. The first set represents transmitting targets individually, and the second through fourth set of targets represent transmitting targets as a consolidated batch. AgileDelta's Chief Technologist John Schneider provided the JRAE 06 data set. Contact information for Mr. Schneider is provided in Appendix A.

c) MIL-STD 6017 Variable Message Format—Operational

The Variable Message Format (VMF) is a joint interoperability standard used to exchange K-series messages between tactical data systems over a number of different tactical data links. Additionally, applications such as AFATDS, C2PC, and others support VMF. The full data set consisted of 55 sensor messages and 64 track messages, which were provided by the Army through AgileDelta. As indicated by the name, the message format is variable, which translates into varying file size. From the original 119 files provided, a subset consisting of two, sensor message formats, K04.20 and K04.24 were selected along with one track message format, K04.1 condition one. AgileDelta's Chief Technologist John Schneider provided the VMF data set. Contact information for Mr. Schneider is provided in Appendix A.

B. NETWORK CONFIGURATION

The ideal environment for testing is the real world using an active deployed ship's network and a live SATCOM connection. Testing was conducted using the Network Technology and Integration Laboratory (NTIL) located at SPAWAR Space Systems Command Pacific in San Diego. The laboratory network consisted of equipment similar to that used by Navy ship and shore installations. Figure 5 illustrates the simulated network architecture, and describes the equipment, software, and version numbers implemented in the testing environment.

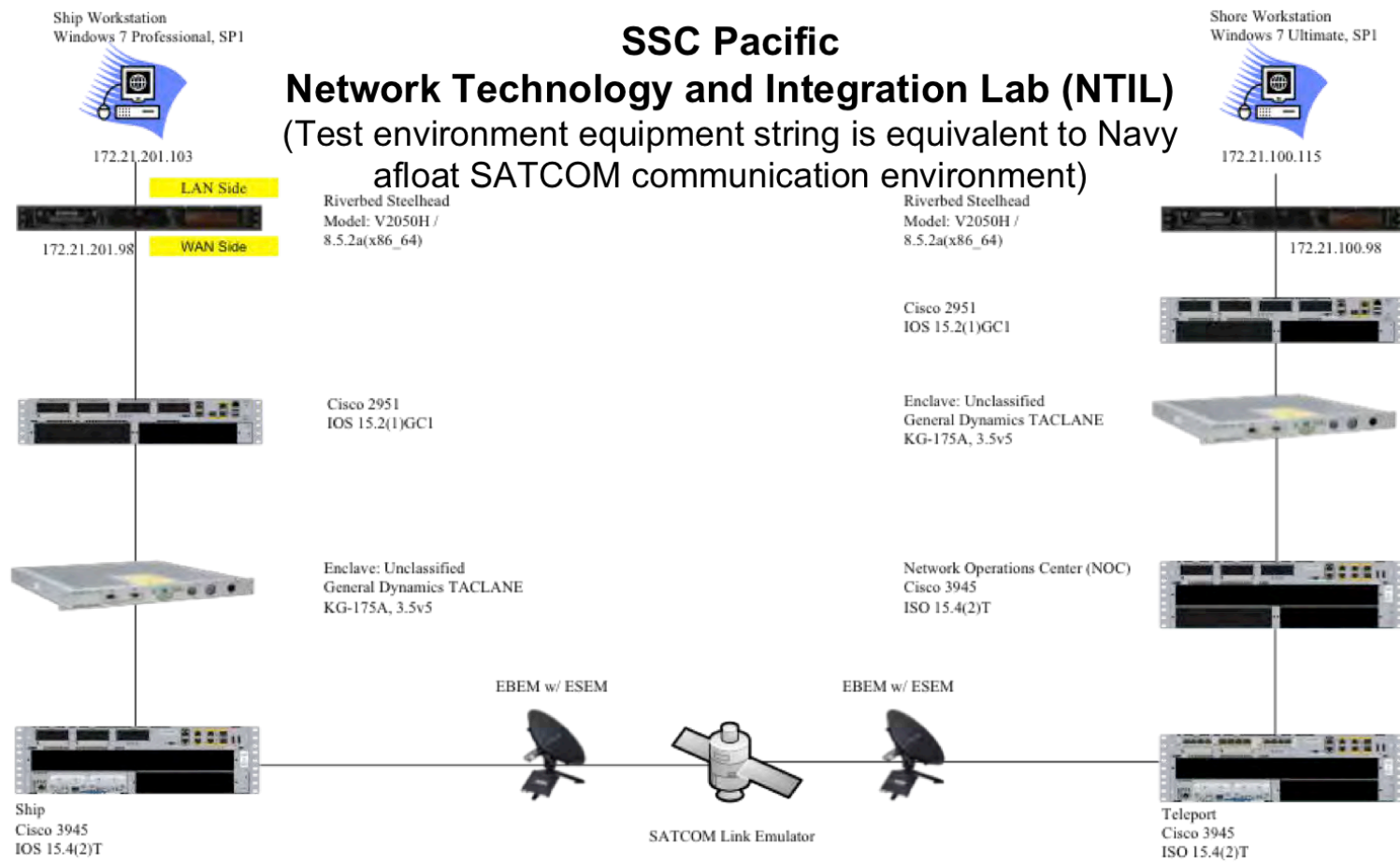


Figure 5. SPAWAR System Center Pacific—Network Technology and Integration Laboratory (NTIL) equipment string.

C. DATA MEASUREMENT AND TRANSFER METHODS

The goal of testing was to identify the change in size of the payload data being exchanged between two systems over the network. The method used to measure payload size dictates the protocol that is best for delivering test data. It must only measure the test data and not include any additional protocol overhead. In order to eliminate any payload overhead associated with protocol operation, the method used to transfer the payload must not introduce any, or at the very least, minimal additional payload. If additional payload is introduced via the data delivery protocol it should be identifiable and mathematically removed from payload calculations during data collection.

The Riverbed Steelhead user interface Current Connections report provides detailed information about established TCP connections and was used to record the LAN side and WAN side payload. The default view of the Current Connections report displays LAN side and WAN side measurements in kilobytes (kB); however, the expanded view depicted in Figure 6 displays the measurements in bytes as well as additional connection information. The additional information displayed includes LAN side and WAN side packets transmitted. These values encompass all TCP packets and are not limited to those carrying payload. Lastly, the expanded view provides the connection type, which can be identified in the corresponding legend, connection age, transport, and notes. The notes entry provides information pertaining to the in-path rule and optimization methods in use.

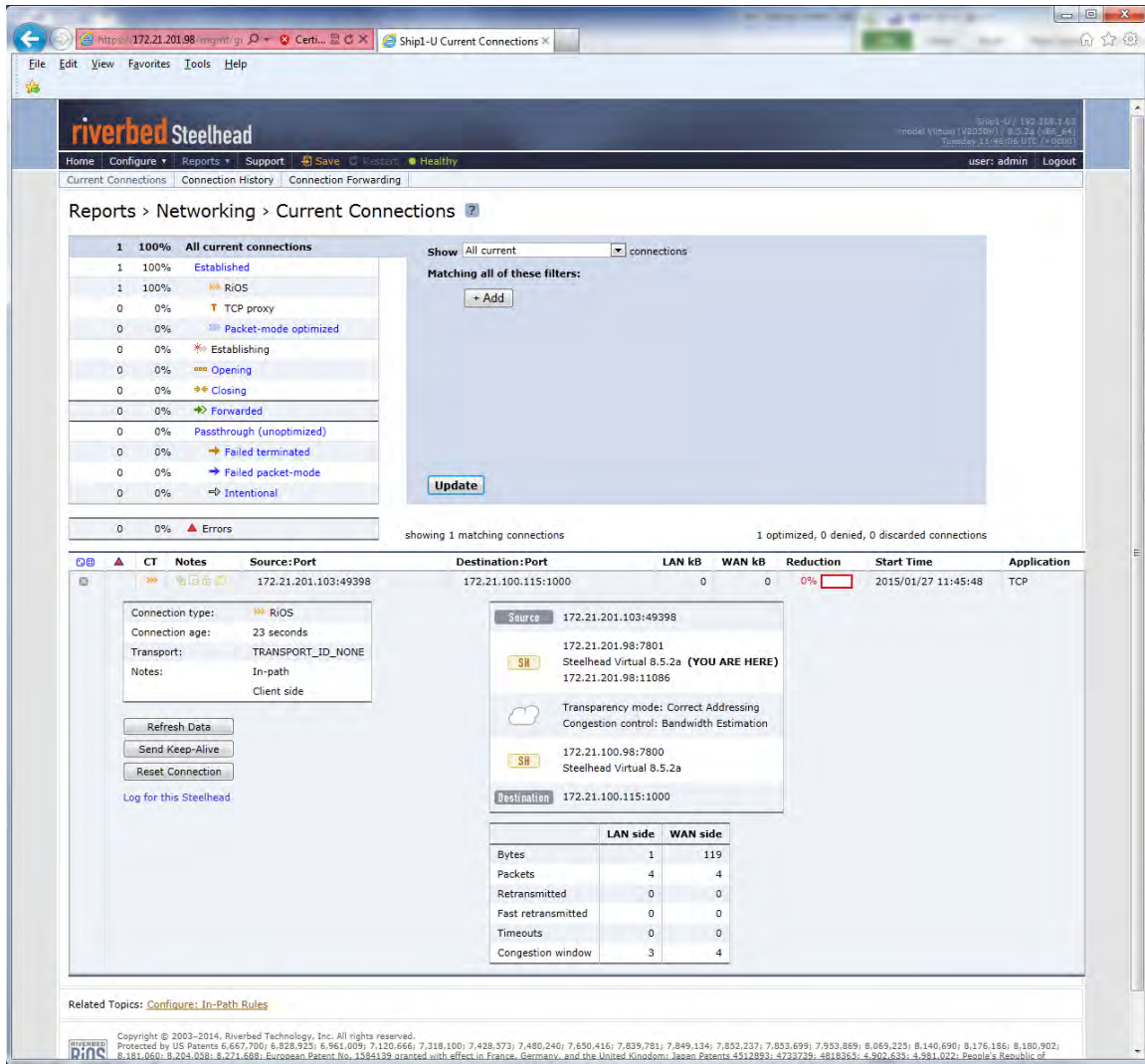
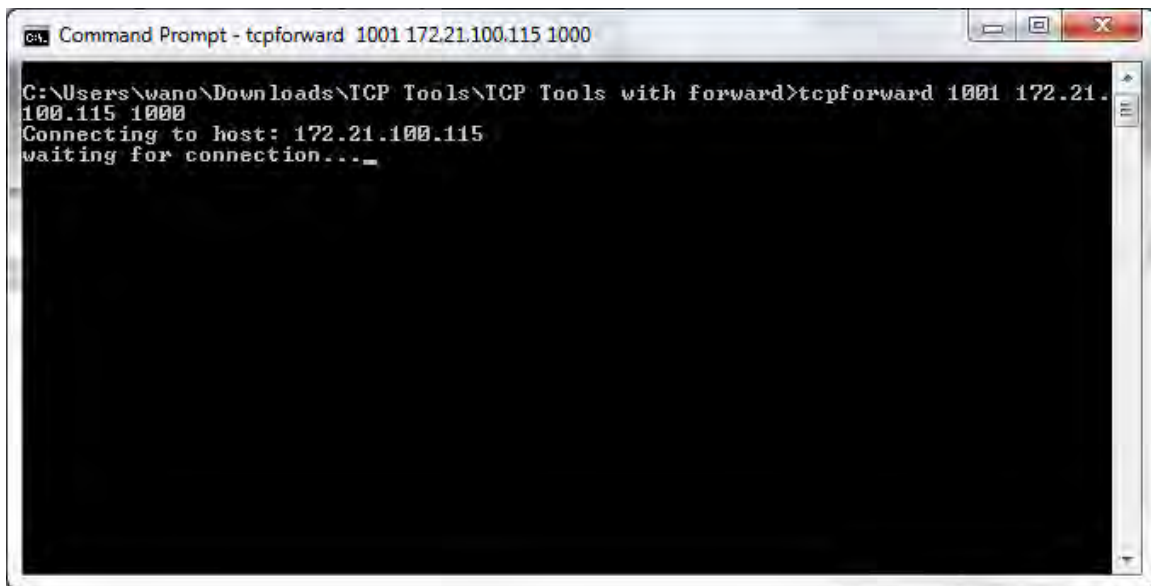


Figure 6. Riverbed Steelhead Current Connections Report, expanded view.

In-path rules are created in order to define what optimization, acceleration, and other features are applied to selected network traffic. Details describing in-path rules used for testing are discussed in subsection D, Features Tested. In order for the Current Connection report details to remain visible to the user after data is transferred, a persistent TCP connection must be established. A persistent TCP connection is one that does not, after the completion of data transfer, send a finish connection packet known as a FIN to terminate the connection. This enables the connection to remain open, meaning that a three-way handshake does not need to occur the next time data is transferred. The

persistent connection results in a cumulative LAN side and WAN side payload measurement that reports the total amount of data transmitted for multiple files.

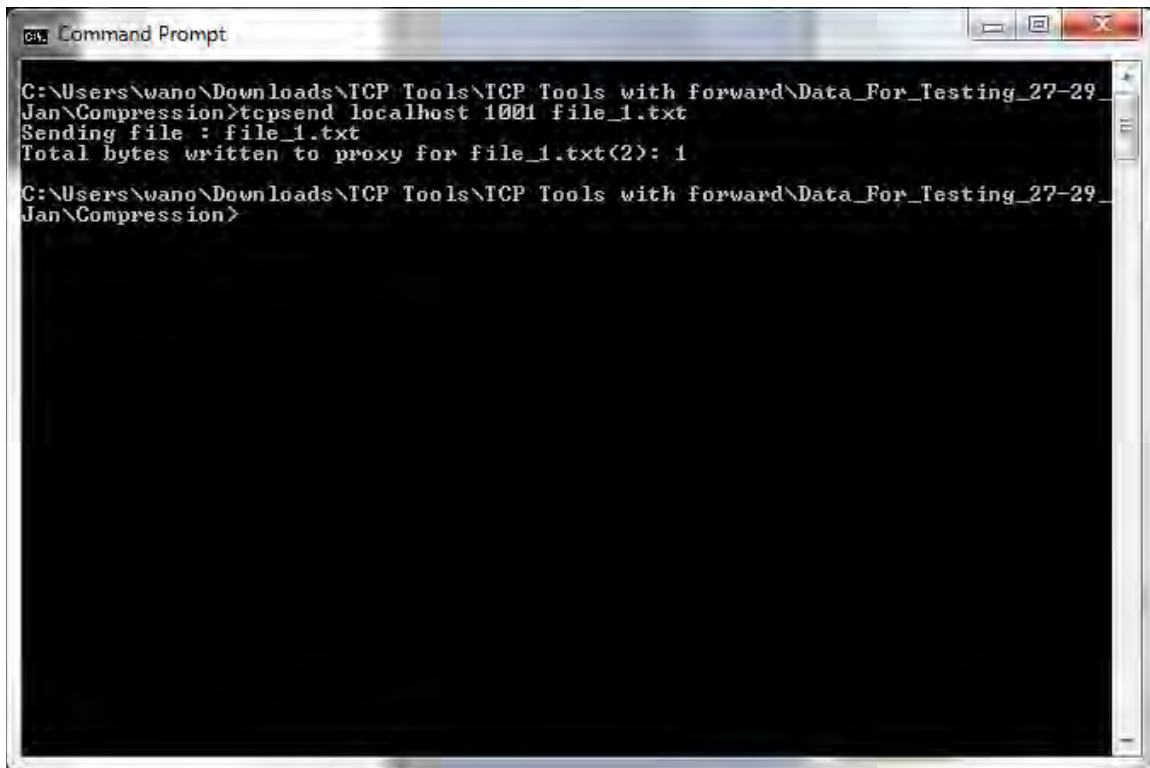
In order to transfer data without incurring any additional application protocol overhead, a command line tool was implemented that transmitted data directly using TCP. The tool used was created by AgileDelta and provided by their Chief Technologist John Schneider. Contact information for Mr. Schneider is provided in Appendix A. The tool is designed using Java TCP sockets and consists of three commands: *tcpforward*, *tcpsend*, and *tcpreceive*. The command line use of *tcpforward*, depicted in Figure 7, is used to establish a persistent TCP connection between the ship workstation with IP address 172.21.201.103 and the shore workstation listening on port 1000 with IP address 172.21.100.115. Once the connection is established, any file sent to *tcpforward*, using port 1001, will be forwarded to the shore workstation, using the persistent connection. The command line use of *tcpsend*, depicted in Figure 8, is used to send files to *tcpforward* using port 1001. Once the file has been successfully transferred, the number of bytes is reflected by *tcpforward* as depicted by the command line screen capture in Figure 9.



```
Command Prompt - tcpforward 1001 172.21.100.115 1000

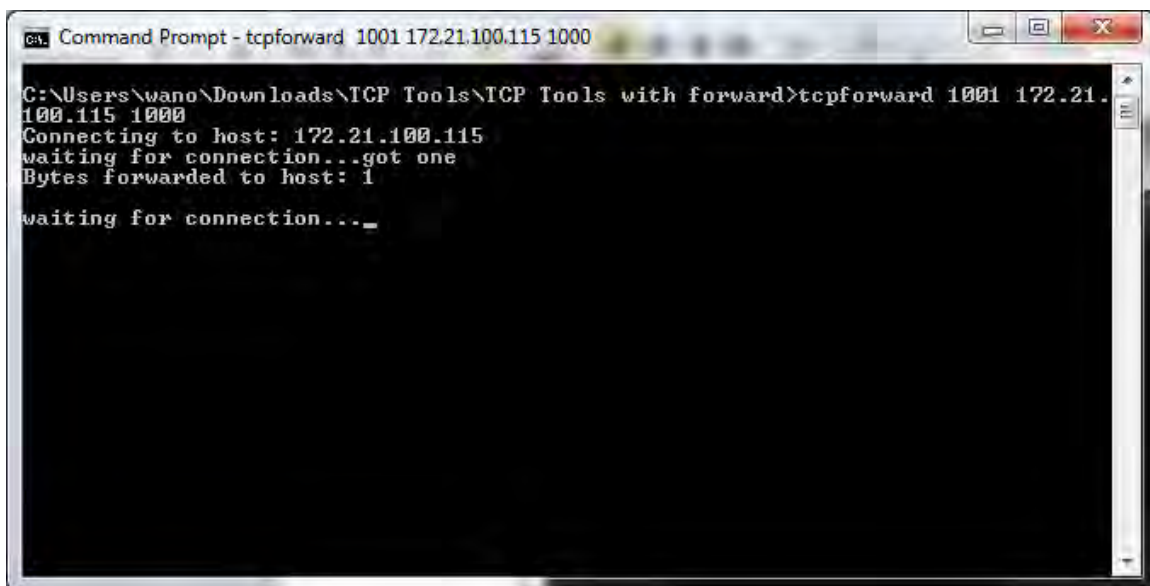
C:\Users\wano\Downloads\TCP Tools\TCP Tools with forward>tcpforward 1001 172.21.100.115 1000
Connecting to host: 172.21.100.115
waiting for connection..._
```

Figure 7. Establishing a persistent connection using *tcpforward*



```
ca. Command Prompt
C:\Users\wano\Downloads\TCP Tools\TCP Tools with forward\Data_For_Testing_27-29_Jan\Compression>tcpsend localhost 1001 file_1.txt
Sending file : file_1.txt
Total bytes written to proxy for file_1.txt(2): 1
C:\Users\wano\Downloads\TCP Tools\TCP Tools with forward\Data_For_Testing_27-29_Jan\Compression>
```

Figure 8. Sending a file using *tcp*send.



```
ca. Command Prompt - tcpforward 1001 172.21.100.115 1000
C:\Users\wano\Downloads\TCP Tools\TCP Tools with forward>tcpforward 1001 172.21.100.115 1000
Connecting to host: 172.21.100.115
waiting for connection...got one
Bytes forwarded to host: 1
waiting for connection..._
```

Figure 9. Successful file transfer reflected by *tcp*forward.

The Riverbed Steelhead includes additional payload in the first packet delivered in order to communicate system configuration parameters in use for the connection. Transmitting a priming file that contained only one character and was one byte in size, identified the amount of connection overhead. Overhead of 118 bytes existed for connections that did not implement LZ Compression or SDR and was calculated by subtracting the number of LAN side bytes from the number of WAN side bytes (119 bytes–1 byte). The screen capture in Figure 10 illustrates the current connection results with no LZ Compression and no SDR. Overhead of 147 bytes depicted in Figures 11–13, existed for connections that implement LZ Compression, SDR, or both LZ Compression and SDR combined.

0

0% ▲ Errors

showing 1 matching connections

1 optimized, 0 denied, 0 discarded connections

	CT	Notes	Source:Port	Destination:Port	LAN kB	WAN kB	Reduction	Start Time	Application
	▶▶▶		172.21.201.103:49179	172.21.100.115:1000	0	0	0% 	2015/01/20 13:00:19	TCP

Connection type:

▶▶▶ RIOS

Connection age:

15 seconds

Transport:

TRANSPORT_ID_NONE

Notes:

In-path

Client side

Refresh Data

Send Keep-Alive

Reset Connection

Log for this Steelhead

Source

172.21.201.103:49179

SH

172.21.201.98:7801

Steelhead Virtual 8.5.2a (YOU ARE HERE)

172.21.201.98:11086

☁

Transparency mode: Correct Addressing

Congestion control: Bandwidth Estimation

SH

172.21.100.98:7800

Steelhead Virtual 8.5.2a

Destination

172.21.100.115:1000

	LAN side	WAN side
Bytes	1	119
Packets	4	4
Retransmitted	0	0
Fast retransmitted	0	0
Timeouts	0	0
Congestion window	3	4

Figure 10. Current connections screen used to identify 118 bytes of overhead for connections using neither LZ Compression or SDR

0

0%

▲ Errors

showing 1 matching connections

1 optimized, 0 denied, 0 discarded connections

CT	Notes	Source:Port	Destination:Port	LAN kB	WAN kB	Reduction	Start Time	Application
		172.21.201.103:49185	172.21.100.115:1000	0	0	0% <div></div>	2015/01/20 13:07:44	TCP

Connection type:

»» RiOS

Connection age:

16 seconds

Transport:

TRANSPORT_ID_NONE

Notes:

In-path

LZ compressed

Client side

Refresh Data

Send Keep-Alive

Reset Connection

Log for this Steelhead

Source

172.21.201.103:49185

SH

172.21.201.98:7801

Steelhead Virtual 8.5.2a (YOU ARE HERE)

172.21.201.98:11087

Transparency mode: Correct Addressing

Congestion control: Bandwidth Estimation

SH

172.21.100.98:7800

Steelhead Virtual 8.5.2a

Destination

172.21.100.115:1000

	LAN side	WAN side
Bytes	1	148
Packets	4	5
Retransmitted	0	0
Fast retransmitted	0	0
Timeouts	0	0
Congestion window	3	4

Figure 11. Current connections screen used to identify 147 bytes of overhead for connections using LZ Compression only

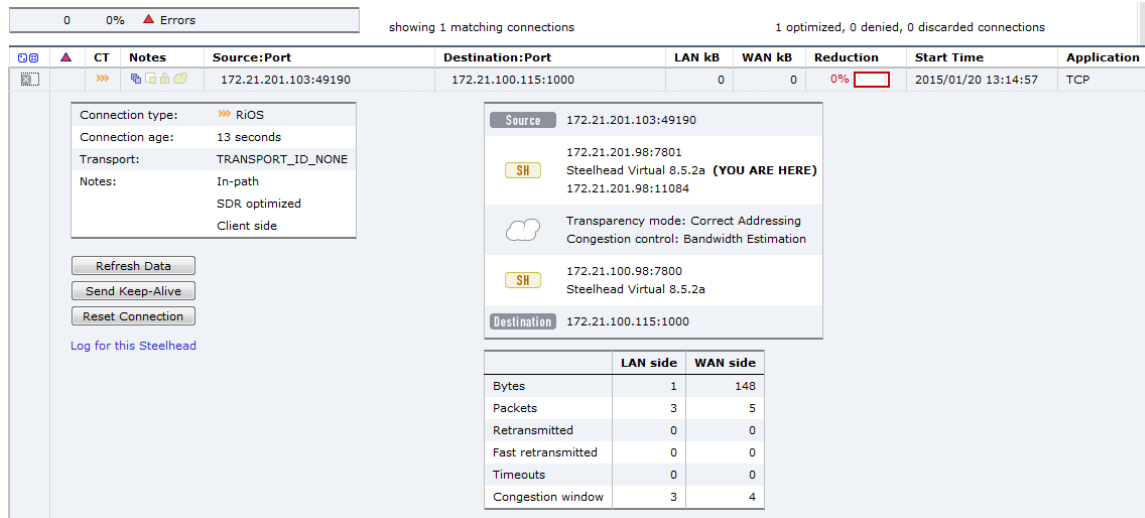


Figure 12. Current connections screen used to identify 147 bytes of overhead for connections using SDR only

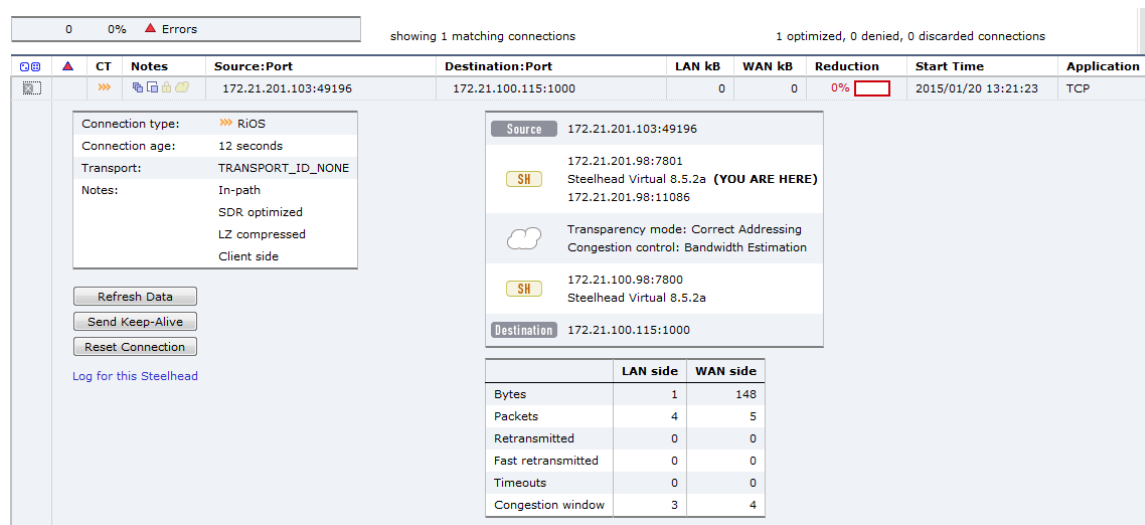


Figure 13. Current connections screen used to identify 148 bytes of overhead for connections using LZ Compression and SDR combined.

A reference baseline was established for the data set and consisted of WAN side values with LZ Compression and SDR disabled. The baseline represents the total WAN payload prior to compression or data deduplication. For all test data, the baseline is equal to the original file size. Details describing testing steps implemented for each optimization feature and specific testing methods, are found in the following subsections.

D. FEATURES TESTED

The following describes and depicts the Riverbed Steelhead configuration and in-path rules used during the testing of each function. Default values for all settings were used where applicable, with the exception of Adaptive Compression, which must be enabled to specifically test the feature.

1. LZ Compression

The goal of this test was to identify only the LZ Compression capabilities of the Riverbed Steelhead. Both XML and EXI formatted test files were transmitted. The reason for transmitting EXI formatted test files is to identify if further compression occurs and if not, what additional overhead may be added to the WAN payload. To isolate the LZ Compression feature, an in-path rule was defined to apply only LZ Compression and, Adaptive Compression was turned off. The screen captures in Figures 14–16 illustrate the configuration menus and selections used. The Riverbed User's Guide describes the levels available for the LZ Compression feature as follows:

Specifies the relative trade-off of data compression for LAN throughput speed. Generally, a lower number provides faster throughput and slightly less data reduction.

Select a RiOS data store compression value of 1 (minimum compression, uses less CPU) through 9 (maximum compression, uses more CPU) from the dropdown list. The default value corresponds to level 6.

Riverbed recommends setting the compression level to 1 in high-throughput environments such as data center-to-data center replication. (Riverbed Technology, 2013b)

riverbed

Steelhead

SRG-11 / 492.186.1.88

model: Virtual (V20009) / R.3.2a (vBR_64)

Thursday, 11/16/03 UTC (+0000)

Home

Configure

Reports

Support

Save

Restart

Healthy

user: admin

Logout

General Service Settings

In-Path Rules

Peering Rules

Transport Settings

Service Ports

Data Store

Performance

Configure > Optimization > In-Path Rules ?

+ Add a New In-Path Rule

- Remove Selected Rules

⇅ Move Selected Rules...

<input type="checkbox"/>	Rule	Type	Source	Destination	VLAN	Protocol	Preoptimization Policy	Latency Policy	Data Reduction Policy	Cloud Acceleration	Kickoff	Status
<input type="checkbox"/>	Q 1	Auto Discover	All-IP:*	All-IP:*	All	--	None	None	None	Auto	No	Disabled
Description: None												
<input type="checkbox"/>	Q 2	Auto Discover	All-IP:*	All-IP:*	All	--	None	None	Compr-Only	Auto	No	Enabled
Description: Compression-Only												
<div>Type: <div>Auto Discover</div></div> <div>Source Subnet: <div>All-IP</div></div> <div>Destination Subnet: <div>All-IP</div></div> <div>VLAN Tag ID: <div>all</div></div> <div>Port or Port Label: <div>all</div></div> <div>Preoptimization Policy: <div>None</div></div> <div>Latency Optimization Policy: <div>None</div></div> <div>Data Reduction Policy: <div>Compression-Only</div></div> <div>Cloud Acceleration: <div>Auto</div></div> <div>Auto Kickoff: <div><input type="checkbox"/></div></div> <div>Neural Framing Mode: <div>Always</div></div> <div>WAN Visibility Mode: <div>Correct Addressing</div></div> <div>Description: <div>Compression-Only</div></div> <div>Enable Rule: <div><input checked="" type="checkbox"/></div></div> <div>Apply</div>												
<input type="checkbox"/>	Q 3	Auto Discover	All-IP:*	All-IP:*	All	--	None	None	SDR-Only	Auto	No	Disabled
Description: SDR-Only												
<input type="checkbox"/>	Q 4	Pass Through	All-IP:*	All-IP:Secure	All	TCP	--	--	--	Auto	--	Enabled
<input type="checkbox"/>	Q 5	Pass Through	All-IP:*	All-IP:Interactive	All	TCP	--	--	--	Auto	--	Enabled
<input type="checkbox"/>	Q 6	Pass Through	All-IP:*	All-IP:RBT-Proto	All	TCP	--	--	--	Auto	--	Enabled
<input type="checkbox"/>	default	Auto Discover	All-IP:*	All-IP:*	All	--	None	Normal	Normal	Auto	No	Enabled
Description: Default In-Path Rule												

Related Topics:

[Configure: General Service Settings](#),
[Configure: Peering Rules](#),
[Configure: SSL Main Settings](#),
[Configure: Certificate Authorities](#),
[Configure: Advanced Settings](#),
[Configure: HTTP](#),
[Configure: MAPI](#),
[Configure: Port Labels](#),
[Reports: Current Connections](#),
[Reports: Connection History](#)

riverbed

POWERED

Copyright © 2003-2014, Riverbed Technology, Inc. All rights reserved.

Protected by US Patents 6,667,700; 6,828,925; 6,961,009; 7,120,666; 7,318,100; 7,428,573; 7,480,240; 7,650,416; 7,839,781; 7,849,134; 7,852,237; 7,853,699; 7,953,869; 8,069,225; 8,140,690; 8,176,186; 8,180,902; 8,181,060; 8,204,038; 8,271,688; European Patent No. 1584139 granted with effect in France, Germany, and the United Kingdom; Japan Patents 4312893; 4733739; 4818363; 4,902,635; 4,981,022; People's Republic of China Patents ZL200380107075.3; ZL200380164909.1; ZL200680013383.X; ZL200680034831.4; Republic of India Patents 219810; 219811; Israel Patent 168274; Hong Kong Patent HK1076935; Patents Pending in the US and other countries.

Figure 14. LZ Compression in-path rule configuration.

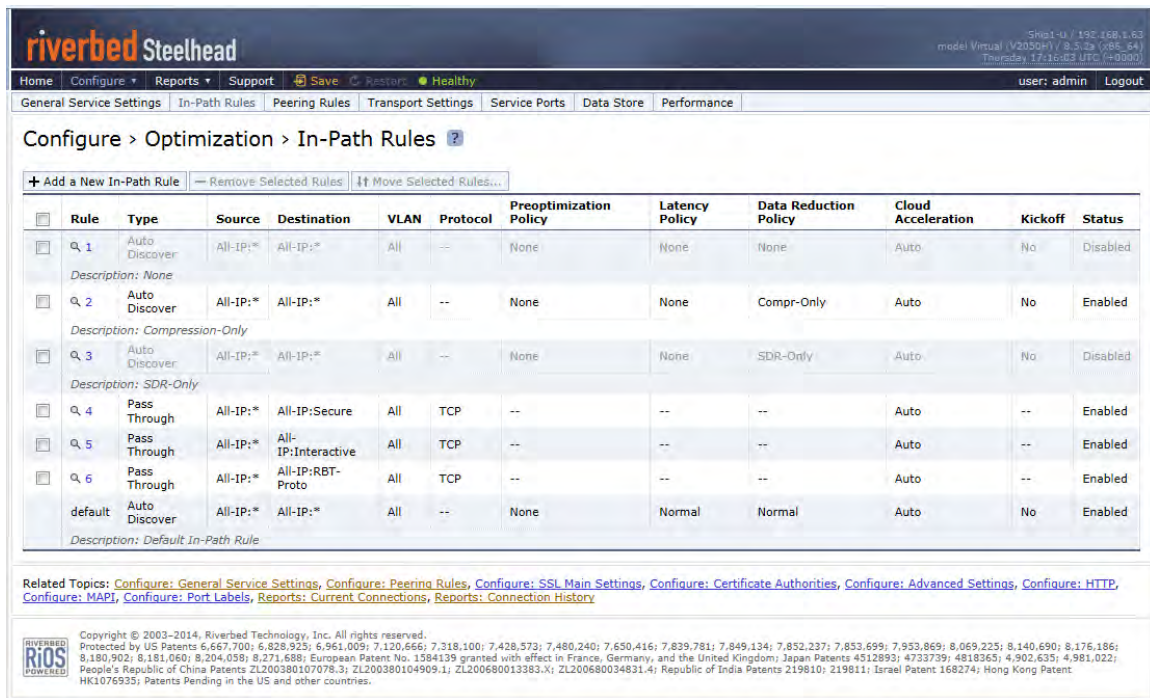


Figure 15. Enabling LZ Compression in-path rule.

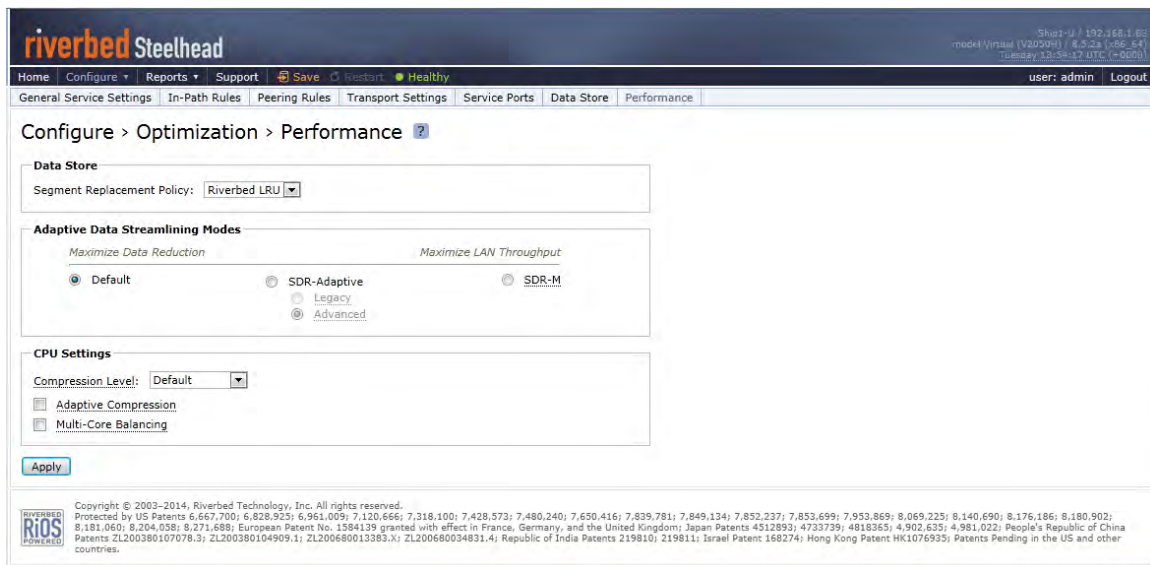


Figure 16. Disabling Adaptive Compression

Data transfers were conducted using LZ Compression Level one, six (default), and nine. Resultant WAN side values were recorded using an Excel data collection spreadsheet.

2. Adaptive Compression

Adaptive Compression is, by default, disabled. The default compression level is 6. The range of compression levels available is 1 through 9. The Riverbed User's Guide describes the Adaptive Compression feature as follows:

Detects LZ data compression performance for a connection dynamically and turns it off (sets the compression level to 0) momentarily if it is not achieving optimal results. Improves end-to-end throughput over the LAN by maximizing the WAN throughput. By default, this setting is disabled. (Riverbed Technology, 2013b)

As stated in Chapter II, the application of LZ compression to previously compressed data can result in an increase in data size. The goal of this test was to identify if WAN payload sizes increased, decreased, or remained the same when compression is the only data reduction policy applied, and the Adaptive Compression feature is enabled. This test utilized the same in-path rule implemented for testing Compression, shown in Figures 14-15, and Adaptive Compression was enabled. The screen capture in Figure 17 illustrate the change in configuration.

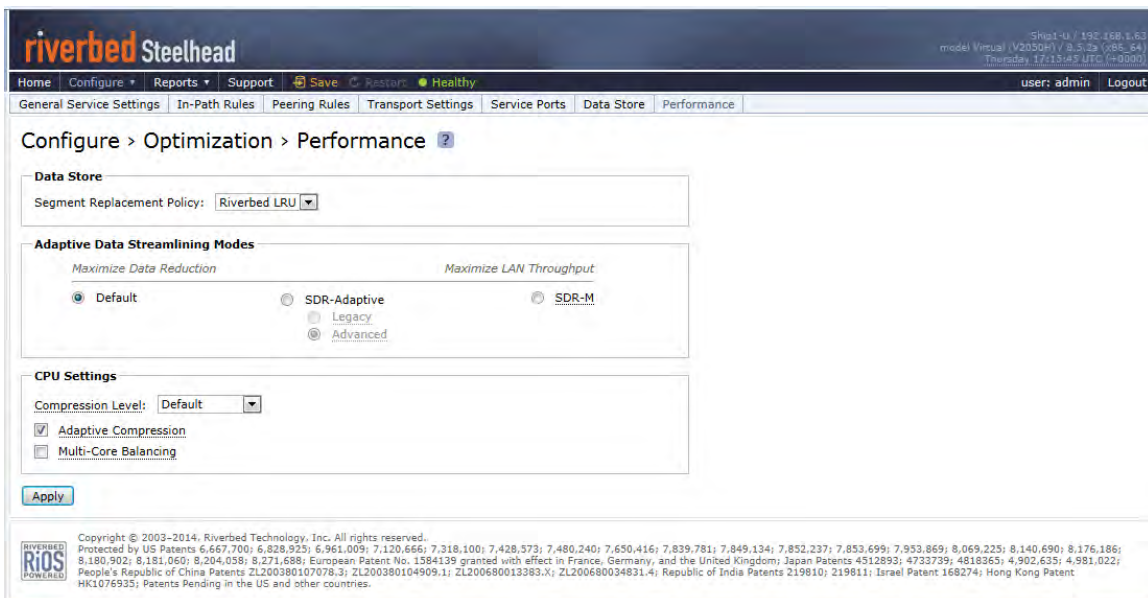


Figure 17. Enabling Adaptive Compression.

Data transfers were conducted using both EXI and Apple Gzip 2 formats with LZ Compression Level one, six (default), and nine. Resultant WAN side values were recorded using an Excel data collection spreadsheet.

3. SDR

When SDR is enabled, additional data store configuration parameters must be set. These parameters include the Data Store Segment Replacement Policy, and the Adaptive Streamlining Mode. Options for the Data Store Segment Replacement Policy are, least recently used (LRU), and first-in, first-out (FIFO). The default configuration is LRU and was the configuration used for all tests conducted in this research. The Adaptive Streamlining Mode provides three modes for operation: Default, SDR-Adaptive, and SDR-M. Each mode provides a balance between throughput and data store read/write operations of varying degree. The SDR-M mode of operation “performs all data reduction in memory, which prevents the Steelhead appliance from reading and writing to and from the disk.”(Riverbed Technology, 2013b, pp. 80–82). While SDR-M can yield high LAN-side throughput by eliminating disk latency, it also reduces the total data store size, thereby limiting the data reduction domain. The screen capture in Figure 18 shows the SDR optimization performance configuration options. An in-path rule was defined to apply only SDR. Screen captures listing the details and application of the in-path rule are shown in Figures 19–20.

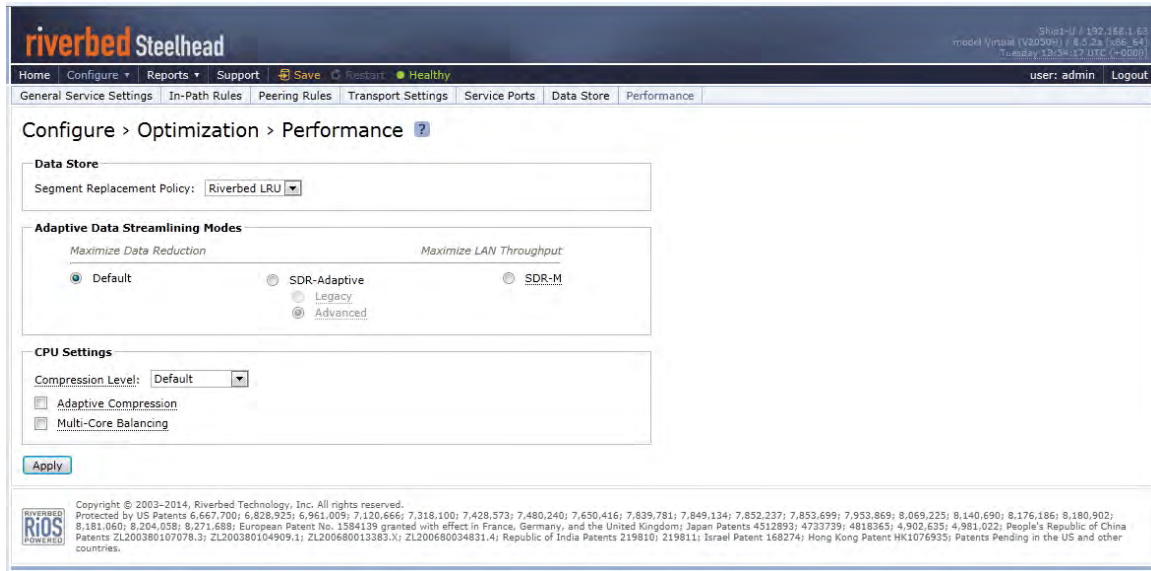


Figure 18. SDR Performance Configuration Options

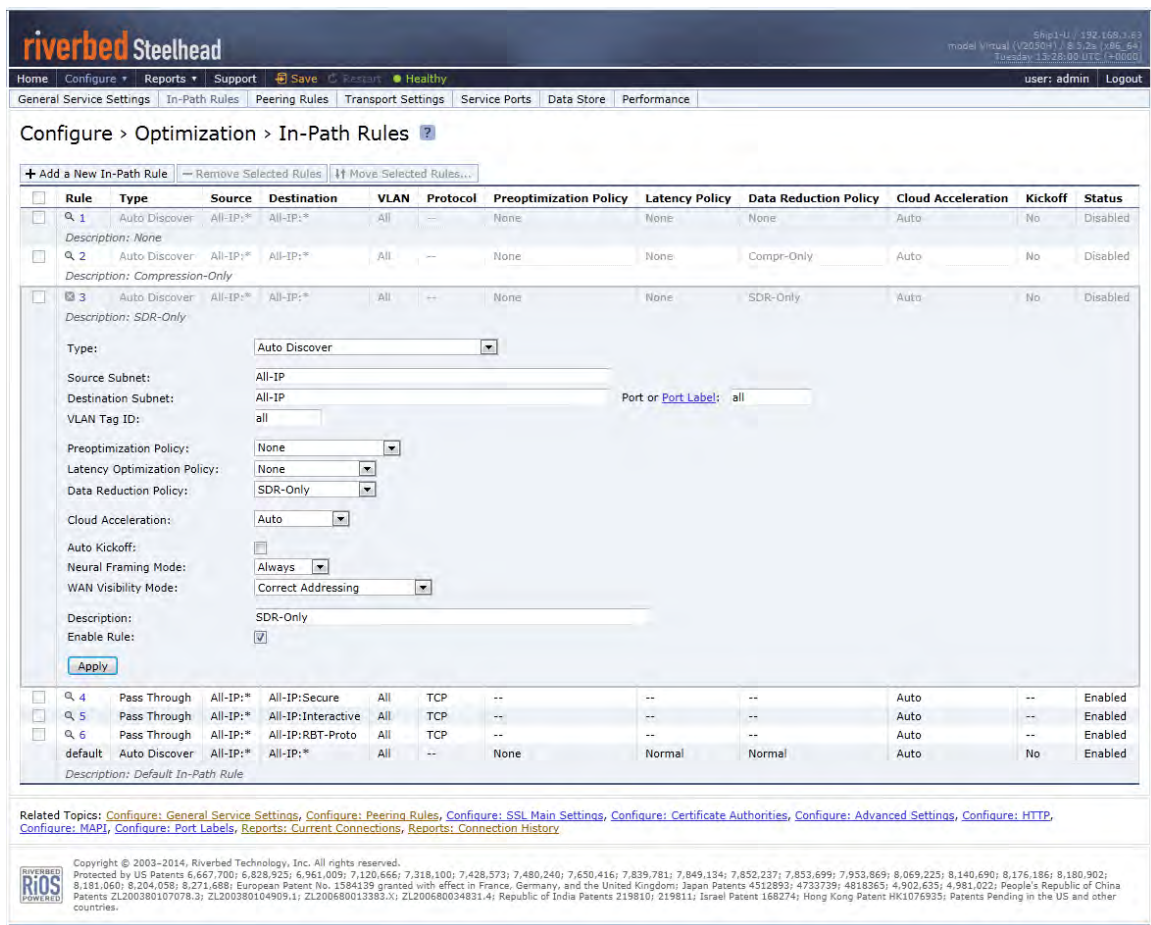


Figure 19. SDR in-path rule configuration.

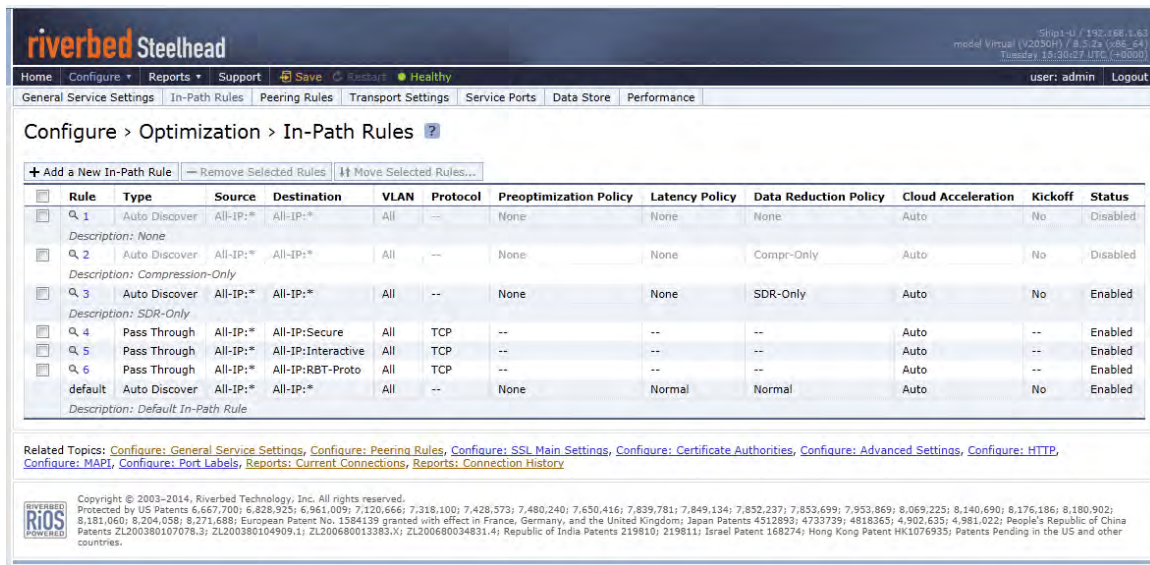


Figure 20. Enabling SDR in-path rule.

When SDR is implemented, two types of transfers can be conducted. The first is a cold transfer, which occurs the first time data is passed through the Riverbed Steelhead. Clearing the data store prior to transmitting data ensures a cold transfer. The second is a warm transfer, which occurs the second time data is passed through the Riverbed Steelhead without clearing the data store between transmissions. With LZ Compression turned off, any change in WAN side payload can be attributed to the SDR function.

a) Identical Files

Identical files for the full data set were transferred across the network five times in succession. Files formatted in XML and EXI were transferred. The WAN payload value of the first file transferred is recorded as the cold transfer result, and the second through fifth WAN payloads are each recorded as warm test results. The primary focus is on the first and second transfers. The third through fifth transfers are used to establish how quickly SDR achieved minimum WAN payload for a given data set. The goal of this test is to identify the payload reduction capability of SDR when presented with identical data. SDR is capable of leveraging data commonalities that exist between applications transmitting data using different protocols. An example of this situation are multiple users browsing to the same website, or sending e-mail with the same files attached. Other tactical applications may apply, however their content are not likely to be exact matches.

b) Single Modification per File

Testing for a single modification per file utilizes only the largest file in the FSM data set. Files formatted in XML and EXI are transferred. Three modifications were made to the original file, one at the beginning, the next in the middle, and the last one at the very end. The end result consists of three files, each containing one modification. The cold transfer consisted of the original file with no modifications. The warm test consisted of one of the modified files. The series of cold and warm transfers were conducted for each of the modified files: beginning, middle, and end. Lastly, the four files were transmitted in succession providing one cold transfer and three warm transfers. The goal of this test was to identify the change in WAN payload achieved using only SDR when a very small change is made to a relatively large amount of data.

c) Multiple Modifications per File

Testing for multiple modifications per file utilizes the Track K04.1 file from the VMF data set. Two versions of the test were executed: (1) the first version changes the values of four different elements within the file, and (2) the second version changes the values of eight different elements within the file. Each test utilized 10 files, each with different element values. A cold transfer is conducted by transferring the first file. The remaining nine files were conducted as warm tests. The goal of this test was to identify the change in WAN payload, achieved using only SDR, when multiple changes are made to element values while holding the files structure constant.

4. LZ Compression and SDR

Testing LZ Compression and SDR combined was conducted using identical test methods used for testing SDR alone, while also enabling LZ Compression at level 6.

E. CHAPTER SUMMARY

This chapter described the test methods used for testing various file formats with the LZ Compression and SDR features of the Riverbed Steelhead. Data collected during the tests was used to generate the graphs and results presented in Chapter IV.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. EXPERIMENTAL RESULTS AND ANALYSIS

This chapter provides a guide for interpreting data, graphs and results of the experiments described in Chapter III. Interpreting the results describes three common methods used for calculating compression ratios and identifies the specific method used by this research. Additionally, details are provided to explain what is meant when a compression method's compactness is described as being greater than another method's. The term compactness is used rather than performance because other variables such as time and memory requirements can influence the overall performance of a compression algorithm. The following results and analysis focus on the reduction of WAN traffic. Graphics and raw data results are provided for each feature tested along with a written analysis answering the respective research questions.

A. INTERPRETING THE RESULTS

There are three common methods for calculating compactness when testing compression capabilities. The method used for the results presented in this research is a compression ratio. The remaining two methods are: percent of original size, and space savings. Calculations for each method are listed below:

- $\text{Compression Ratio} = \text{uncompressed size} / \text{compressed size}$
- $\text{Percent of Original Size} = \text{compressed size} / \text{original size}$
- $\text{Space Savings} = 1 - (\text{compressed size} / \text{original size})$

The compression ratio is a ratio between the uncompressed size and the compressed size of data. Compression ratios are often annotated for example, 10:1 for a ratio calculated using values 100/10, and spoken as "10 times smaller than the original size." Graphing compression ratio values provides a uniform Y-axis with greater compactness depicted by larger values, and lesser compactness depicted by smaller values. The maximum value of the Y-axis is determined by the greatest compactness achieved for the given data set. Where drastic compactness differences exist, smaller compression ratios are depicted with their differences being less discernable, while larger

compression ratios are depicted with their differences being more discernable. Compression ratio was selected as the value for graphing because it clearly shows the extent of differences between results of greater compactness.

The percent of original size calculation is the inverse of a compression ratio. Graphing percent of original size values also provides a uniform Y-axis but greater compactness is depicted by smaller values. The maximum value of the Y-axis is determined by the lowest compactness achieved for the given data set, and the greatest compactness appears at the bottom of the graph. Values can be displayed as a decimal, with the value 1 indicating no compression or as a percentage with 100% indicating no compression.

The space savings calculation is a relative calculation showing size reduction relative to original data size. Space savings values are often annotated as a percentage. Results depicted by space savings do not clearly show the differences between results. The example data and calculations in Table 2 list the results associated with compression ratio and space savings calculations.

Table 2. Example calculations comparing compression ratio and space savings calculations. Results can be misleading if not carefully considered.

Compression Ratio vs Space Savings				
Data Set			Compression Ratio	Space Savings
	Uncompressed	Compressed	Original / Compressed	1 - (Compressed / Original)
Set 1	1,000	10	100	99.00%
Set 2	10,000	10	1,000	99.90%

The compression ratio achieved with data set #2 is 10 times greater than the compression ratio achieved with data set #1. To determine by what magnitude one compression ratio outperforms another, simply divide the larger ratio by the smaller ratio. For the example data used above: $1,000 / 100 = 10$. The compression applied to data 2 is described as 10 times greater than that applied to data set #1. Only a .9% increase occurs due to the magnitude 10 improvement in data set #2 when calculated using the space

savings equation. The small change in percentage causes moderate compression to appear nearly equal to one that is 10 times greater. The value presented for “Reduction” on the Riverbed Steelhead Current Connections Report uses the space savings formula, and was therefore not utilized in this research to determine Riverbed Steelhead LZ Compression or SDR performance.

The terms and abbreviations listed in Table 3 are used in the legends of the graphs and analysis used in the rest of this chapter.

Table 3. Terms and abbreviations used for graph legends and analysis.

Abbreviation / Term	Definition
SH	Riverbed Steelhead WAN Optimization Controller is referred to as the <i>Steelhead</i> or abbreviated SH.
Comp	The compression feature of the SH is abbreviated Comp
Adaptive	Indicates that Adaptive Compression was turned on. The default setting is to have Adaptive Compression turned off. All compression was conducted with Adaptive Compression turned off unless specifically annotated.
SDR	The Scalable Data Referencing feature is abbreviated SDR.
SDR Cold	Indicates the first instance of data being transferred through the SH. Prior to conducting cold transfers, the SH data store was reset in order to ensure no deduplication was possible based on previous test data submissions.
SDR Warm	Indicates the second iteration of data being transferred through the SH. Warm transfers are any transfer occurring after a cold transfer. The data transferred can range from an exact copy of the first file transferred, to a file with one or more difference from the first.

XML + SH Comp	XML data with SH LZ Compression applied to it. Unless identified differently, the original format of the data used is XML.
XML + SDR	XML data with SH SDR applied to it.
Gzip	XML data with Gzip applied to it.
EXI	XML data encoded as EXI.

B. COMPRESSION—RESEARCH QUESTIONS 1 AND 2

The following graphs, data, and analysis are in response to research questions #1 and #2. Research question #1 asks: Can EXI provides greater compression than Riverbed Steelhead's LZ Compression for XML data? Research question #2 asks: Does compactness using Riverbed Steelhead's LZ Compression vary with compression level?

EXI provides greater compression than SH Comp for all three data sets. The graph in Figure 21 shows the contrast between EXI formatted data and XML data compressed using SH Comp. The graph in Figure 22 shows the compression factor gained by using EXI over SH Comp. The values listed in Table 4 are the file sizes in bytes of the uncompressed, XML + SH Comp Level 6, and EXI formats of the files included in the three data test sets. The file sizes were used to calculate the compression ratios listed in Table 5. The values listed in Table 5 are the compression ratios and improvement factors for the files included in the three data test sets. Compression ratios achieved by SH Comp range from 2.06:1 to 16.28. Compression ratios achieved by EXI range from 10.08:1 to 118.90. At a minimum EXI encoding achieves 1.61 times greater compactness over SH Comp and at most 19.38 times greater compactness over SH Comp.

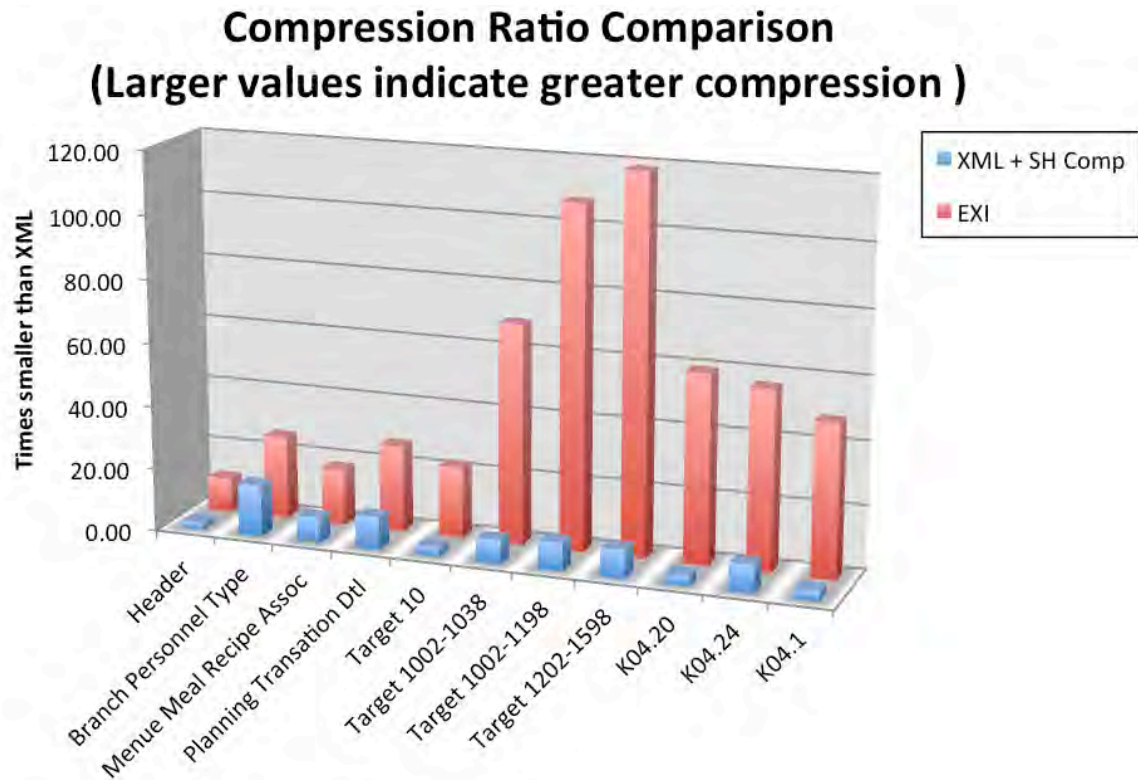


Figure 21. Graph showing compression ratio comparisons for FSM, JRAE and VMF data sets. Larger Y-axis values indicate greater compression.

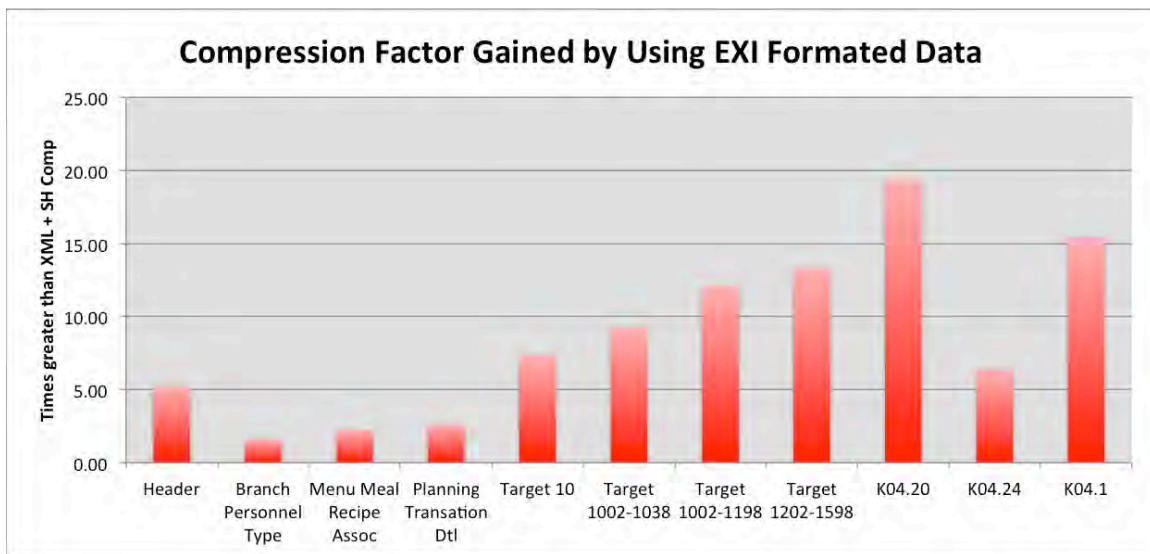


Figure 22. Graph showing additional compression factor gained by using EXI over SH Comp for FSM, JRAE, and VMF data sets. The values shown indicate by how many times better EXI compression is than SH Comp.

Table 4. Data table listing original file size and resulting file size following SH Comp and EXI for FSM, JRAE, and VMF data sets.

Original and Compressed File Sizes (all data values listed in bytes)			
File Name	XML Size	XML + SH Comp	EXI
Header	648	315	60
Branch Personnel Type	26,701	1,640	1,016
Menu Meal Recipe Assoc	68,786,829	8,655,759	3,855,746
Planning Transation Dtl	291,223,416	27,855,906	10,679,035
Target 10	2,718	892	121
Target 1002-1038	26,228	3,500	378
Target 1002-1198	129,992	14,576	1,204
Target 1202-1598	260,037	29,083	2,187
K04.20	2,685	872	45
K04.24	30,127	3,396	532
K04.1	2,948	943	61

Table 5. Data table listing compression ratios achieved using SH Comp, EXI, and corresponding improvement factors for FSM, JRAE, and VMF data sets.

Compression Ratio Improvement Comparison			
File Name	SH Comp	EXI	Improvement factor
Header	2.06	10.80	5.25
Branch Personnel Type	16.28	26.28	1.61
Menu Meal Recipe Assoc	7.95	17.84	2.24
Planning Transation Dtl	10.45	27.27	2.61
Target 10	3.05	22.46	7.37
Target 1002-1038	7.49	69.39	9.26
Target 1002-1198	8.92	107.97	12.11
Target 1202-1598	8.94	118.90	13.30
K04.20	3.08	59.67	19.38
K04.24	8.87	56.63	6.38
K04.1	3.13	48.33	15.46

The graphs in Figures 23–25 show the contrast between SH Comp ratios achieved by levels 1, 6, and 9 for the files in each data set. SH Comp compactness varied slightly between levels 1, 6, and 9. Variations between levels 6 and 9 are negligible. Level 6 is the default compression level and was used for all other testing. The values listed in Table 6 are the compression ratios achieved for the files included in the three data test sets using SH Comp level 1, 6, and 9.

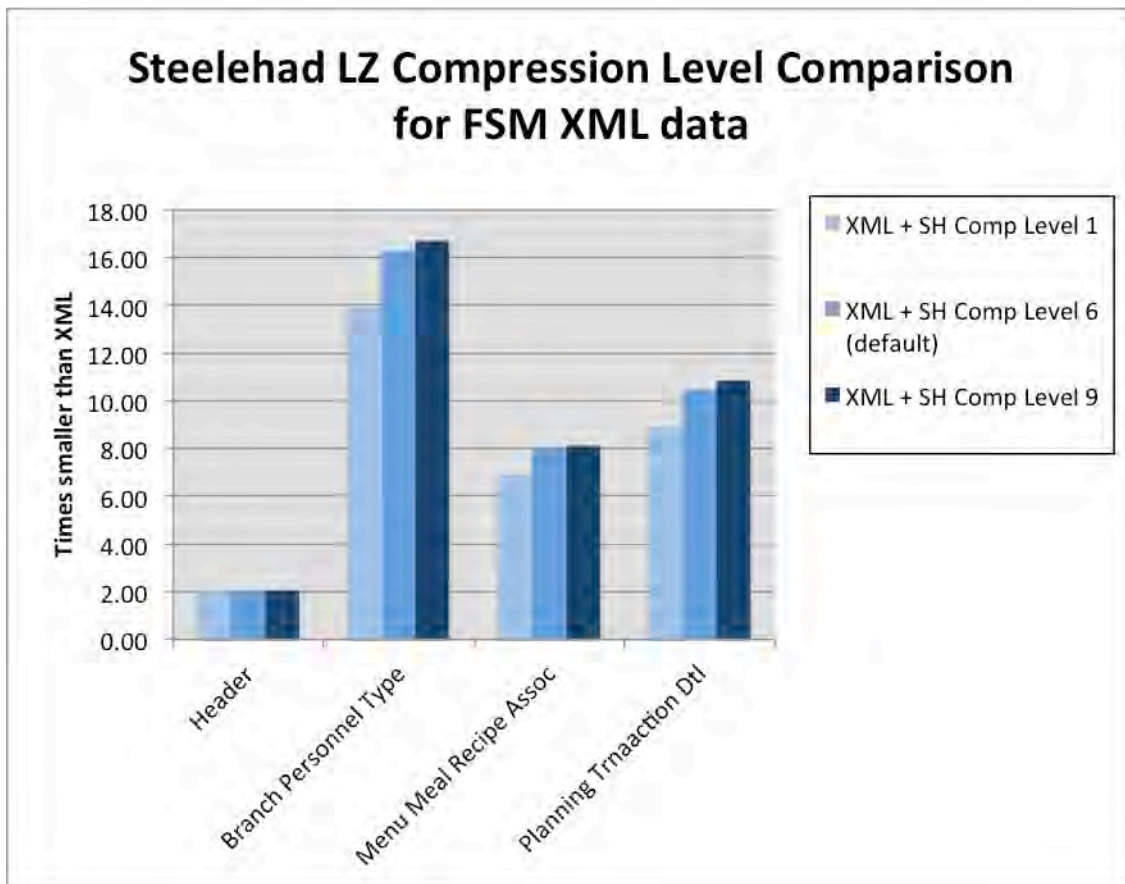


Figure 23. Graph comparing Steelhead compression levels 1,6, and 9 for the FSM data set.

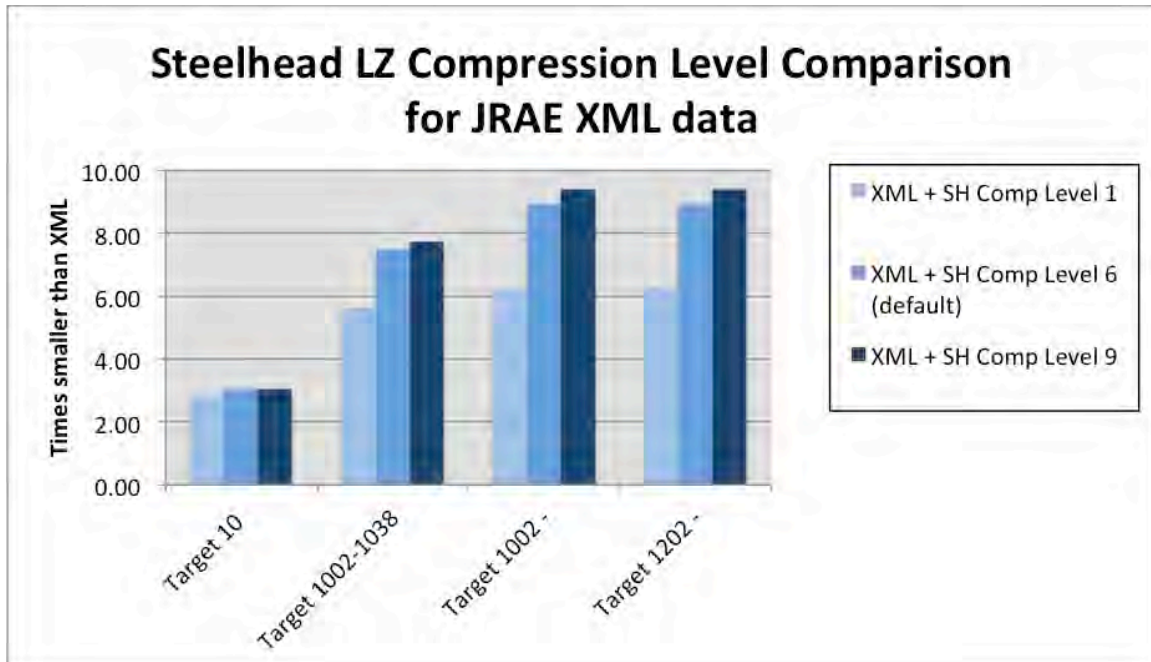


Figure 24. Graph comparing Steelhead compression levels 1,6, and 9 for the JRAE data set.

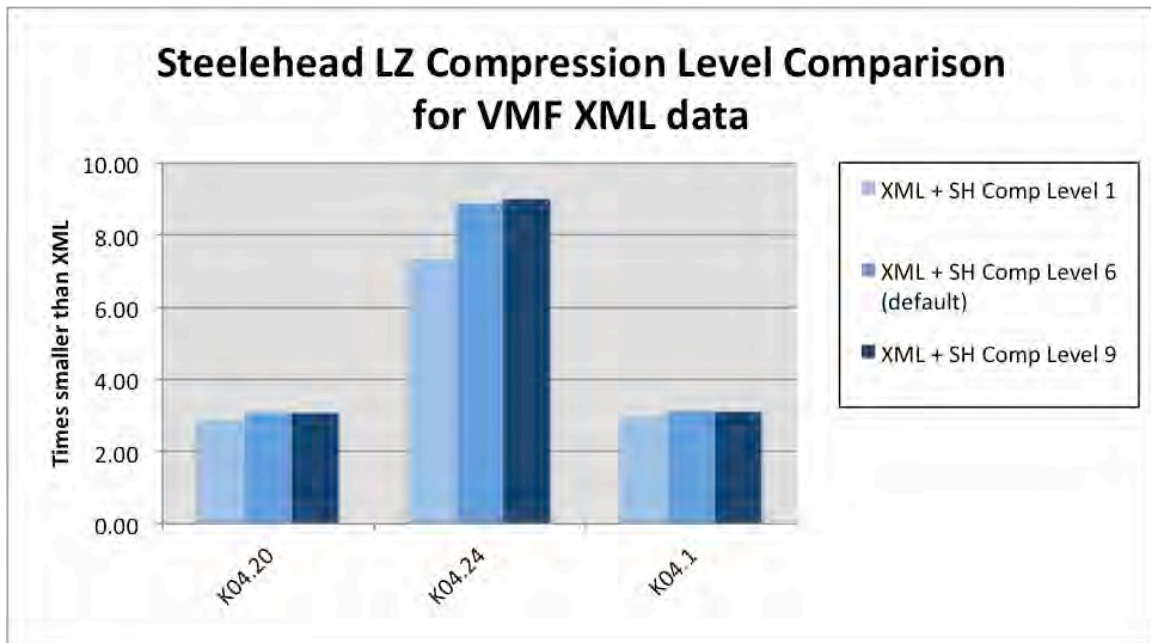


Figure 25. Graph comparing Steelhead compression levels 1,6, and 9 for the VMF data set.

Table 6. Data table showing compression ratios achieved using Steelhead LZ Compression levels 1, 6, and 9 for FSM, JRAE, and VMF data sets.

Level 1, 6, and 9 Compression Ratio Values			
File Name	Level 1	Level 6 (default)	Level 9
Header	1.98	2.06	2.06
Branch Personnel Type	13.90	16.28	16.70
Menu Meal Recipe Assoc	6.87	7.95	8.12
Planning Transation Dtl	8.91	10.45	10.82
Target 10	2.83	3.05	3.05
Target 1002-1038	5.64	7.49	7.75
Target 1002-1198	6.23	8.92	9.40
Target 1202-1598	6.26	8.94	9.41
K04.20	2.87	3.08	3.08
K04.24	7.35	8.87	9.02
K04.1	2.98	3.13	3.12

C. ADAPTIVE COMPRESSION—RESEARCH QUESTION 3

The following data, and analysis are in response to research question #3. Research question #3 asks: What effect does Riverbed Steelhead's Adaptive Compression have on Gzip and EXI compressed data?

Enabling the Riverbed Steelhead's Adaptive Compression feature does not eliminate the increase in payload size caused by the application of SH Comp to previously compressed data. The increase in overall size of the data transmitted over the WAN results in the decrease of total compactness for that connection. The severity of impact caused by the second application of compression is influenced by the size of the previously compressed data. Previously compressed data that is small in relation to the amount of overhead added, by a second compression attempt, suffers a greater reduction in effective compactness. Due to EXI compressed data being significantly smaller than its Gzip counterpart, effective EXI compactness is diminished more. The values in Table 7 show the size in bytes associated with the files in each data set before and after the application of SH Comp. In addition, Table 7 includes the change in size, with a positive value indicating an increase in size and a negative value indicating decrease in size.

Table 7. Data table listing original file size, resulting file size, and the difference following the application of Steelhead Adaptive Compression to Gzip and EXI data for FSM, JRAE, and VMF data sets.

Adaptive Compression Level 6 Raw Data (all data values listed in bytes)								
File Name	XML Size	XML + SH Comp	Gzip	Gzip + SH Adaptive	Difference	EXI	EXI + SH Adaptive	Difference
Header	648	315	320	360	40	60	89	29
Branch Personnel Type	26,701	1,640	1,661	1,701	40	1,016	1,060	44
Menu Meal Recipe Assoc	68,786,829	8,655,759	8,424,827	8,426,064	1,237	3,855,746	3,857,582	1,836
Planning Transation Dtl	291,223,416	27,855,906	26,988,298	27,000,543	12,245	10,679,035	10,605,890	-73,145
Target 10	2,718	892	892	936	44	121	165	44
Target 1002-1038	26,228	3,500	3,507	3,551	44	378	422	44
Target 1002-1198	129,992	14,576	14,584	14,633	49	1,204	1,248	44
Target 1202-1598	260,037	29,083	28,352	28,411	59	2,187	2,231	44
K04.20	2,685	872	883	927	44	45	74	29
K04.24	30,127	3,396	3,408	3,452	44	532	576	44
K04.1	2,948	943	954	998	44	61	90	29

All test cases except one grew in size. The largest test case “Planning Transaction Dtl”, from the FSM data set benefited from the second application of LZ based compression applied through SH Comp. This deviation is possibly attributed to the need for further schema refinement. It must be noted that further compression was achieved only after the initial application of EXI. A second application of LZ based compression applied to the previously Gzipped version of the file does not receive any additional benefit but rather increases in size. This further supports that additional compression achieved is due to compressibility remaining after EXI encoding and not necessarily due to the LZ based compression applied through SH Comp. The amount of overhead added to test files with sizes ranging from a few hundred bytes to hundreds of kilobytes is around 44 bytes, regardless of file format (XML or EXI). Smaller EXI encoded files receive less overhead (29 bytes) and larger file sizes in the megabytes receive additional overhead.

The difference between the Gzip compression ratio values shown in Table 8 and the SH Comp ratio values shown in Table 6 are not significantly different. As answered by research question 1, compactness of EXI encoding exceeds that achieved by SH Comp in all test cases. The impact caused by the overhead associated with double compression is shown in Table 8 by listing the ratios achieved before and after the application of SH Comp. The compression achieved in smaller test files is diminished slightly more than in

the larger test files. For a system transferring a mix of small and large files, the overall reduction may be negligible.

In order to achieve maximum compression and avoid any diminished results, SH Comp can be turned off using an in-path rule, for any data transmissions known to contain EXI or other compressed data. If the detailed knowledge of network traffic, required for an in-path rule, is not available, EXI compactness will still exceed that of LZ based compression, after the application of SH Comp. Therefore EXI is the preferred compression method for structured, semi-structured data that may be formatted as XML.

Table 8. Raw data table showing the change in compression ratios for Gzip and EXI after the application of Steelhead Adaptive Compression.

Adaptive Compression Comparison (ratios before and after)				
File Name	Gzip	Gzip + SH Adaptive	EXI	EXI + SH Adaptive
Header	2.03	1.80	10.80	7.28
Branch Personnel Type	16.08	15.70	26.28	25.19
Menu Meal Recipe Assoc	8.16	8.16	17.84	17.83
Planning Transation Dtl	10.79	10.79	27.27	27.46
Target 10	3.05	2.90	22.46	16.47
Target 1002-1038	7.48	7.39	69.39	62.15
Target 1002-1198	8.91	8.88	107.97	104.16
Target 1202-1598	9.17	9.15	118.90	116.56
K04.20	3.04	2.90	59.67	36.28
K04.24	8.84	8.73	56.63	52.30
K04.1	3.09	2.95	48.33	32.76

D. COMPRESSION + SDR COLD TRANSFER—RESEARCH QUESTION 4

The following graphs, data, and analysis are in response to research question #4. Research question #4 asks: What effect does Riverbed Steelhead’s LZ Compression + SDR have on XML data and EXI compressed data when transmitted as a cold transfer?

Data deduplication is not designed to provide significant reduction for data transmitted during a cold transfer. When transferring data for the first time, the creation of data store segments and the associated segment references slightly increases the total amount of data sent across the WAN. The increase in total data transferred is essentially a “price of admission” necessary to achieve the data reduction associated with a warm

transfer. The values in Table 9 show the size in bytes associated with the files in each data set before and after the application of SDR for a cold transfer. In addition, Table 9 includes the change in size, indicated by a positive value. Smaller files incurred less overhead than larger files. This is expected because larger files will consists of a larger number of segments and therefore a larger number of segment references will need transferred over the WAN to the peer system. Because smaller files incur less overhead, the EXI encoded data naturally receives less overhead than XML compressed using LZ Compression.

The highlighted values in Tables 9 and 10 draw attention to an inconsistent application of SDR to test files whose EXI compressed file size is 121 bytes or less. In these test cases, it appears that no additional overhead associated with SDR occurred and therefore it is assumed that segment references were either not created or at a minimum were not transferred over the WAN. Without the creation and transmission of segment references, the benefits of SDR during a warm transfer will not be possible. While the largest file size affected in the data set was 121 bytes, the upper file size threshold associated with this anomaly is unclear. The file size values listed in Table 9 indicate that file sizes greater than or equal to 378 bytes have segment references created and transferred across the WAN. The four file names not expected to receive SDR benefits during a warm transfer are Header, Target10, K04.20, and K04.1.

Table 9. Data table listing original file size, resulting cold transfer file size, and the difference following the application of Steelhead's SDR to previously compressed data for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.

Compression + Cold SDR Raw Data (all data values listed in bytes)								
File Name	XML Size	XML + SH Comp	SDR Cold	Difference	EXI	EXI + SH Comp	EXI + SH Comp + SDR Cold	Difference
Header	648	315	364	49	60	89	89	0
Branch Personnel Type	26,701	1,640	2,293	653	1,016	1,056	1,109	53
Menu Meal Recipe Assoc	68,786,829	8,655,759	10,561,458	1,905,699	3,855,746	3,714,662	3,805,990	91,328
Planning Transation Dtl	291,223,416	27,855,906	31,109,097	3,253,191	10,679,035	10,389,378	10,644,814	255,436
Target 10	2,718	892	963	71	121	161	161	0
Target 1002-1038	26,228	3,500	4,114	614	378	418	469	51
Target 1002-1198	129,992	14,576	16,653	2,077	1,204	1,244	1,305	61
Target 1202-1598	260,037	29,083	34,192	5,109	2,187	2,207	2,284	77
K04.20	2,685	872	935	63	45	74	74	0
K04.24	30,127	3,396	3,929	533	532	572	619	47
K04.1	2,948	943	1,008	65	61	90	90	0

Table 10. Data table listing original compression ratio and resulting SDR cold transfer compression ratio for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.

SDR Cold Transfer Compression Ratio Comparison (values listed are compression ratios)					
File Name	SH Comp	SH Comp + SDR Cold	EXI	EXI + SH Comp	EXI + SH Comp + SDR Cold
Header	2.06	1.78	10.80	7.28	7.28
Branch Personnel Type	16.28	11.64	26.28	25.29	24.08
Menu Meal Recipe Assoc	7.95	6.51	17.84	18.52	18.07
Planning Transation Dtl	10.45	9.36	27.27	28.03	27.36
Target 10	3.05	2.82	22.46	16.88	16.88
Target 1002-1038	7.49	6.38	69.39	62.75	55.92
Target 1002-1198	8.92	7.81	107.97	104.50	99.61
Target 1202-1598	8.94	7.61	118.90	117.82	113.85
K04.20	3.08	2.87	59.67	36.28	36.28
K04.24	8.87	7.67	56.63	52.67	48.67
K04.1	3.13	2.92	48.33	32.76	32.76

E. COMPRESSION + SDR WARM TRANSFER—RESEARCH QUESTION 5

The following graphs, data, and analysis are in response to research question #5. Research question #5 asks: What effect does Riverbed Steelhead's LZ Compression + SDR have on XML data and EXI compressed data when transmitted as a warm transfer?

SDR provides greater warm transfer compression for EXI encoded data than XML formatted data. SDR achieves the greatest benefit by sending segment references in place of the full segment for data that is a duplicate of any data previously processed by the Steelhead. The graph in Figure 26 shows the contrast between SDR warm compression for EXI formatted data and XML formatted data. The graph in Figure 27 shows the compression factor gained by using EXI encoded data over XML formatted data. The warm transfer consisted of an exact copy of a previously transmitted file. As identified by the results of research question four, the files: Header, Target10, K04.20, and K04.1 did not receive any benefits from warm SDR because segments and segment references were not created during the cold transfer. The WAN payload sizes listed in Table 11 for the files not receiving SDR remained constant following the application of SH Comp + SDR Cold. The constant WAN payload size is reflected by an improvement factor value of one, listed in Table 12. For all other test cases, EXI formatted data experienced greater warm transfer compression than XML formatted data. Compression

ratios achieved using XML formatted data range from 14.73:1 to 1,238.02:1. Compression ratios achieved using EXI encoded data range from 596.09:1 to 7,165.23. At a minimum EXI SDR warm transfers achieve a 1.32 times improvement factor over XML SDR warm transfers and at most a 19.05 times improvement factor over XML SDR warm transfers.

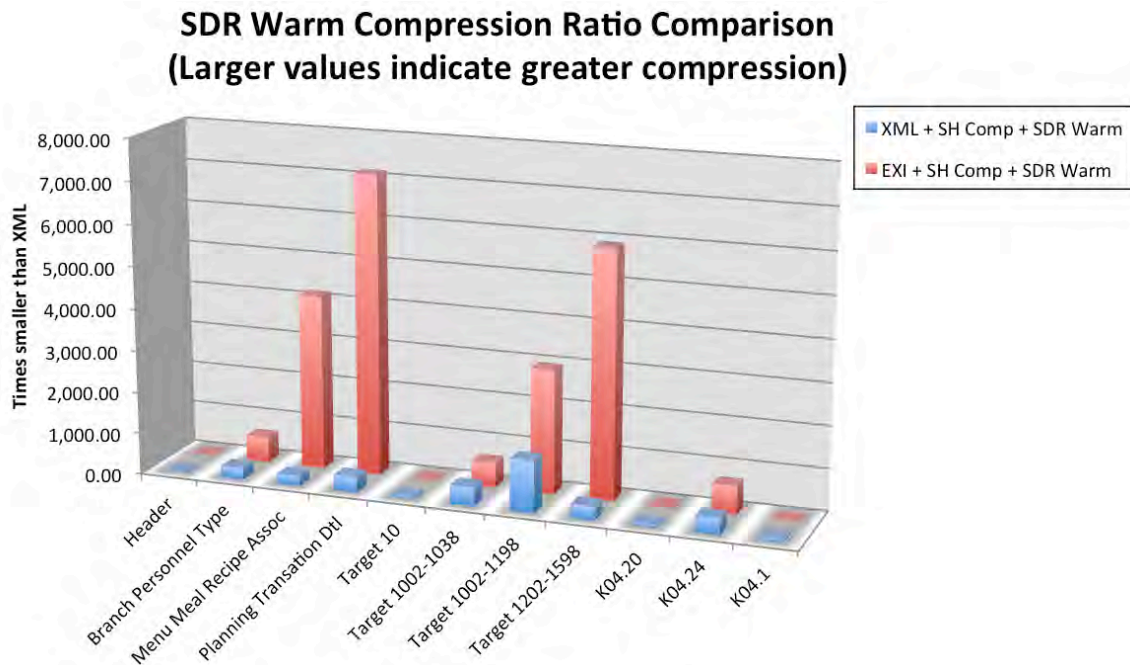


Figure 26. Graph showing SDR warm compression ratio comparisons for FSM, JRAE and VMF data sets. Larger Y-axis values indicate greater compression.

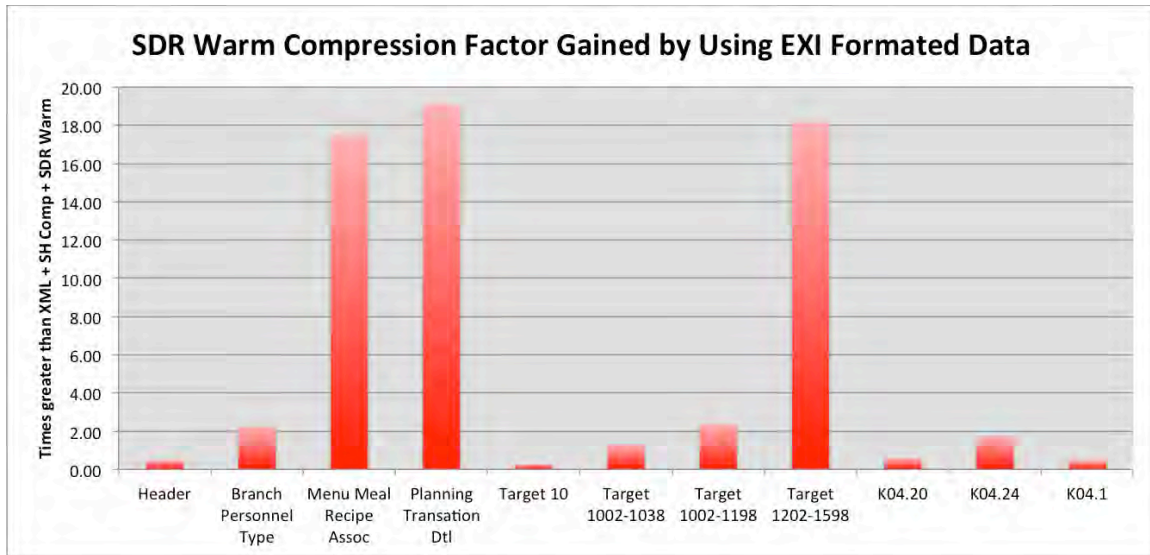


Figure 27. Graph showing additional compression factor gained by using EXI over XML for FSM, JRAE, and VMF data sets. The values shown indicate by how many times better EXI compression is than SH Comp.

Table 11. Data table listing original file size, resulting SDR cold and warm transfer file sizes for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.

Compression + Cold & Warm SDR Raw Data (all values in bytes)								
File Name	XML Size	XML + SH Comp	XML + SH Comp + SDR Cold	XML + SH Comp + SDR Warm	EXI	EXI + SH Comp	EXI + SH Comp + SDR Cold	EXI + SH Comp + SDR Warm
Header	648	315	364	44	60	89	89	89
Branch Personnel Type	26,701	1,640	2,293	98	1,016	1,056	1,109	44
Menu Meal Recipe Assoc	68,786,829	8,655,759	10,561,458	287,805	3,855,746	3,714,662	3,805,990	16,474
Planning Transation Dtl	291,223,416	27,855,906	31,109,097	774,155	10,679,035	10,389,378	10,644,814	40,644
Target 10	2,718	892	963	44	121	161	161	161
Target 1002-1038	26,228	3,500	4,114	58	378	418	469	44
Target 1002-1198	129,992	14,576	16,653	105	1,204	1,244	1,305	44
Target 1202-1598	260,037	29,083	34,192	798	2,187	2,207	2,284	44
K04.20	2,685	872	935	44	45	74	74	74
K04.24	30,127	3,396	3,929	78	532	572	619	44
K04.1	2,948	943	1,008	44	61	90	90	90

Table 12. Data table listing original compression ratio and resulting SDR cold and warm transfer compression ratio for FSM, JRAE, and VMF data sets. Highlighted values indicate where SDR does not appear to have been applied, even when enabled.

SDR Warm Transfer Performance Comparison (values listed are compression ratios and performance % difference)								
File Name	SH Comp	SH Comp + SDR Warm	Improvement Factor		EXI	EXI + SH Comp	EXI + SH Comp + SDR Warm	Improvement Factor
Header	2.06	14.73	7.16	10.80	7.28	7.28	7.28	1.00
Branch Personnel Type	16.28	272.46	16.73	26.28	25.29		606.84	24.00
Menu Meal Recipe Assoc	7.95	239.00	30.08	17.84	18.52		4,175.48	225.49
Planning Transation Dtl	10.45	376.18	35.98	27.27	28.03		7,165.23	255.62
Target 10	3.05	61.77	20.27	22.46	16.88		16.88	1.00
Target 1002-1038	7.49	452.21	60.34	69.39	62.75		596.09	9.50
Target 1002-1198	8.92	1,238.02	138.82	107.97	104.50		2,954.36	28.27
Target 1202-1598	8.94	325.86	36.44	118.90	117.82		5,909.93	50.16
K04.20	3.08	61.02	19.82	59.67	36.28		36.28	1.00
K04.24	8.87	386.24	43.54	56.63	52.67		684.70	13.00
K04.1	3.13	67.00	21.43	48.33	32.76		32.76	1.00

F. SDR EFFECT ON A SINGLE MODIFICATION AT BEGINNING, MIDDLE, END—RESEARCH QUESTIONS 6 AND 7

The following graphs, data, and analysis are in response to research questions #6 and #7. Research question #6 asks: What effect does Riverbed Steelhead's SDR have on XML data and EXI compressed data when a single modification is made at the beginning, middle, and end of a file? Research question #7 asks: What effect does Riverbed Steelhead's LZ Compression + SDR have on XML data and EXI compressed data when a single modification is made at the beginning, middle, and end of a file?

The results of this test support those previously found in that SDR provides greater warm transfer compression for EXI encoded data than XML formatted data. The effects of warm transfer compression achieved through SDR are shown in the graphs contained in Figures 28 and 29. The graph in Figure 30 shows the performance gained by using EXI encoded data over XML formatted data. The graph in Figure 28 illustrates the impact of a single modification occurring at the beginning, middle, and end of a file when the original file is a cold transfer and each modified file represents an isolated warm transfer. Files were first transferred individually in order to isolate the warm transfer compression associated with each modified file. Isolating SDR performance to a single warm transfer required the Steelhead data store to be reset after each modified file was transferred. Resetting the data store should occur very infrequently in an operational environment. For both EXI encoded data and XML formatted data, a file modified in the middle did not achieve the same level of compression as those modified at the beginning

or end. The compression ratios achieved by EXI encoded data and displayed in Figure 28 cause the variations experienced by XML formatted data, while not appreciable, to be slightly obscured. The WAN payload sizes in bytes and compression ratios listed in Tables 13 and 14 respectively, better illustrate the variations in compression achieved by XML formatted and EXI encoded data.

As mentioned, resting the data store between transfers does not represent how the Steelhead would be utilized in an operational environment. A more realistic use of the data store and file transfer is to transfer the original file followed by the modified files without resetting the data store between transfers. Results from transfers conducted in succession are shown in Figure 29. The first modified file transferred after the original file, receives warm transfer compression benefits. The second and third modified files each receive additional warm transfer compression benefits beyond those of the first. Warm transfer compression achieved for the second and third modified files were nearly identical. Using EXI encoded data, the difference between the warm transfer compression achieved for the second and third modified files more than doubled.

The last comparison conducted is between the warm transfer compression achieved for EXI encoded data with the addition of SH Comp and without. The compression ratio and improvement factors listed in Table 14 show that the addition of SH Comp to EXI encoded data produces negligible difference in overall compression.

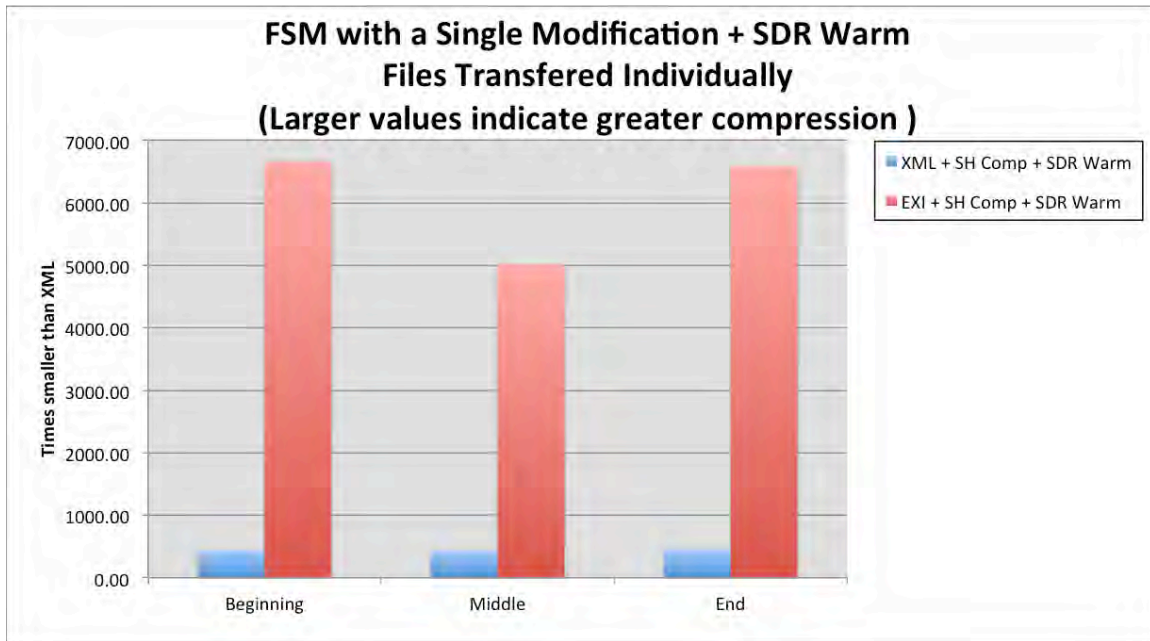


Figure 28. Graph for FSM file “Planning Transaction Dtl” with modifications occurring at the beginning, middle, and end of the file. Files transferred individually.

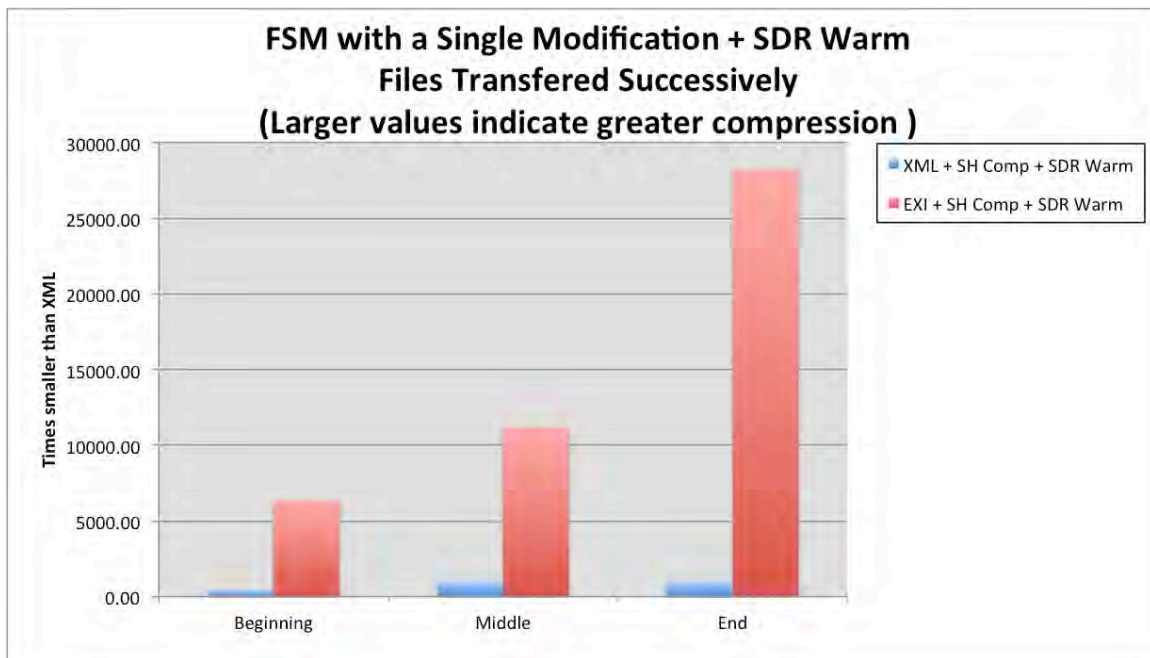


Figure 29. Graph for FSM file “Planning Transaction Dtl” with modifications occurring at the beginning, middle, and end of the file. Files transferred in succession.

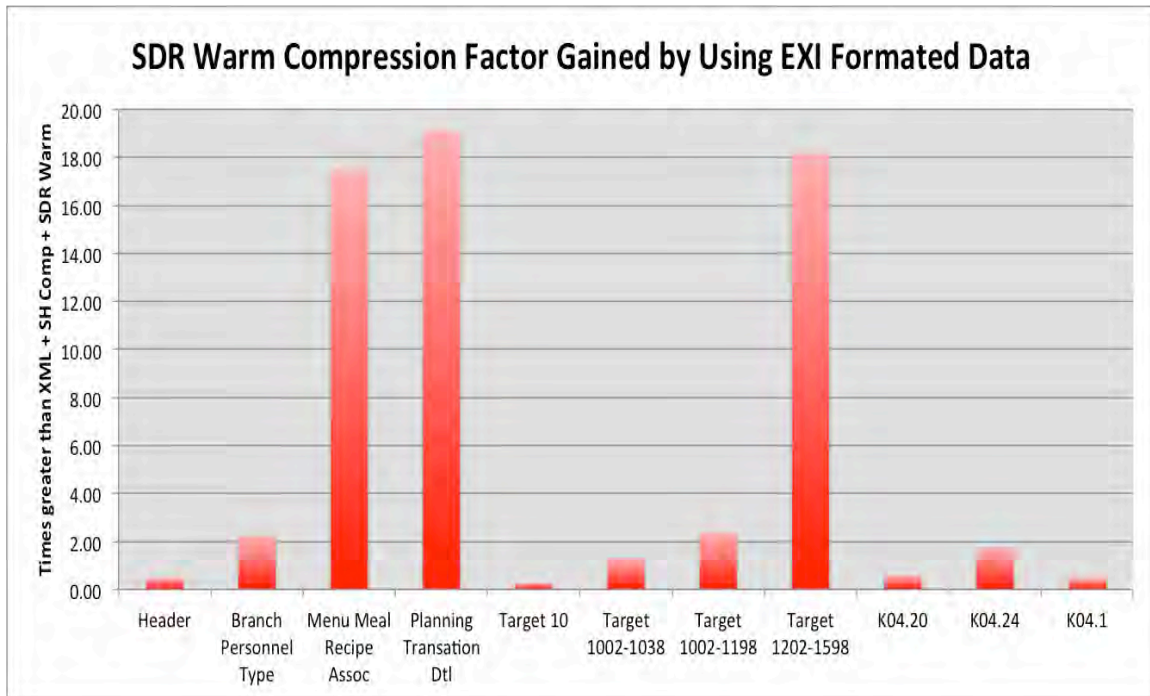


Figure 30. EXI performance improvement over XML for SDR warm transfers containing a modification at the beginning, middle, and end of the file.

Table 13. Data table showing original file size and payload sizes for FSM file “Planning Transaction Dtl” with modifications occurring at the beginning, middle, and end of the file. Includes data for files transferred individually and in succession.

FSM with a single Modification Raw Data (all values in bytes)						
File Name	Modification	XML Size	XML + SDR Warm	XML + SH Comp + SDR Warm	EXI + SDR Warm	EXI + SH Comp + SDR Warm
Files Transferred Individually						
Planning Transaction Dtl	Original	291,223,416	N/A	N/A	N/A	N/A
	Beginning	291,223,415	1,960,670	649,317	42,747	43,716
	Middle	291,223,417	1,939,876	654,036	58,084	58,066
	End	291,223,417	1,960,337	617,918	45,068	44,243
Files Transferred Successively						
Planning Transaction Dtl	Original	291,223,416	N/A	N/A	N/A	N/A
	Beginning	291,223,415	1,960,689	625,337	44,807	45,671
	Middle	291,223,417	278,817	280,189	25,799	26,023
	End	291,223,417	315,602	278,724	10,314	10,317

Table 14. Data table showing compression ratios for FSM file “Planning Transaction Dtl” with modifications occurring at the beginning, middle, and end of the file. Includes ratios for files transferred individually and in succession. Additionally an improvement factor is shown illustrating negligible difference in overall compression when adding SH Comp to EXI.

FSM with a single Modification Compression Ratios							
File Name	Modification	XML + SH			EXI + SH		
		XML + SDR	Comp + SDR	Improvement	Comp + SDR	Improvement	
		Warm	Warm	Factor	Warm	Factor	
Files Transferred Individually							
Planning Transaction Dtl	Original	N/A	N/A	N/A	N/A	N/A	N/A
	Beginning	149	449	6,813	15.19	6,662	14.85
	Middle	150	445	5,014	11.26	5,015	11.26
	End	149	471	6,462	13.71	6,582	13.97
Files Transferred Successively							
Planning Transaction Dtl	Original	N/A	N/A	N/A	N/A	N/A	N/A
	Beginning	149	466	6,500	13.96	6,377	13.69
	Middle	1,044	1,039	11,288	10.86	11,191	10.77
	End	923	1,045	28,236	27.02	28,228	27.02

G. SDR EFFECT ON GREATER THAN ONE MODIFICATION—RESEARCH QUESTIONS 8 AND 9

The following analysis is in response to research questions #8 and #9. Research question #8 asks: What effect does Riverbed Steelhead’s SDR have on XML data and EXI compressed data containing more than one modification between files? Research question #9 asks: What effect does Riverbed Steelhead’s LZ Compression + SDR have on XML data and EXI compressed data containing more than one modification between files?

As described in Chapter III, under heading E, subsection 3.c), testing the effect of SDR on files containing multiple modifications utilized the Track K04.1 file from the VMF data set. As identified in the answers to research questions four and five, SDR was not actually applied to files less than 121 bytes in size. Unfortunately the Track K04.1 file is less than 121 bytes and did not receive SDR warm transfer compression benefits. Therefore the tests conducted in support of this research question are voided due to SDR not having actually been applied when the files were transferred. It is theorized based on the test results already presented that warm transfer compression ratios would range from those achieved using identical files to those achieved by a cold transfer. The actual amount of warm data transfer compression achieved would depend on the degree of

commonality between the file transferred and any file prior to it. Rather than attempt to artificially generate additional test data for this scenario, it is recommended that live network traffic be monitored to identify the actual benefits achieved.

H. CHAPTER SUMMARY

Experimental test results show that for all cold data transfer test cases EXI performance exceeds the performance provided by Riverbed Steelhead's LZ Compression. The use of EXI encoded data over XML formatted data was shown to provide greater warm data transfer compression achieved through Riverbed Steelhead's SDR capability. Riverbed Steelhead's Adaptive Compression feature unfortunately did not eliminate the additional overhead incurred by applying LZ compression to previously compressed data. In tests where EXI encoded data was used and Riverbed's LZ Compression was enabled, the LZ Compression slightly diminished the compactness of EXI alone. Taking the small decrease into consideration, EXI compactness still exceeds LZ Compression. Additionally Riverbed Steelhead's SDR feature did not provide benefits for files transferred with a size of 121 bytes or less. The small file anomaly associated with SDR impacted four files in the tests associated with questions four and five, and voided the results for questions eight and nine. Based on the results and analysis discussed in this chapter, the role of EXI in Navy WAN optimization, along with areas requiring further research is identified in Chapter V.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

EXI was found to provide significant performance improvements over current WOC capabilities when implemented in a complementary manner. The reader is reminded that the author's goal when conducting this research was not to identify whether to use EXI or a WOC, but rather how the two technologies can be best used together. This chapter draws conclusions about the role of EXI in Navy WAN optimization, identifies friction points faced by the Navy with implementing EXI, and recommends further work needed to achieve the greatest overall performance for Navy communication systems.

A. THE ROLE OF EXI IN NAVY WAN OPTIMIZATION

The research conducted shows that EXI is the preferred data encoding and compression method for use with structured and semi-structured data sources, which can be formatted and transmitted as XML data. The price of admission as it is called in this research associated with SDR is well worth the cold data transfer cost based on the excellent performance improvements provided to warm data transfers. While XML formatted data benefited from SDR, the same information encoded using EXI receives substantially greater benefit. There were no test cases where SDR did not provide benefits. For those reasons it is recommended that SDR or equivalent deduplication algorithm be applied for all data transmitted.

1. Limitations Imposed by Encryption

As EXI encoded data replaces XML formatted data for network traffic that is not encrypted prior to arriving at the Riverbed Steelhead, no action is needed on the part of network administrators to achieve the additional SDR benefits associated with the EXI encoded data. It is important to remember that EXI is applied at the Application layer and therefore still achieves its peak compression regardless of encryption applied by lower layers of the TCP/IP protocol stack. For traffic that is encrypted using SSL/TLS, network owners should consider configuring the Riverbed Steelhead's SSL/TLS acceleration to enable LZ Compression and SDR benefits for that traffic. Identifying SSL/TLS traffic

flows over Navy networks is an area currently requiring additional research. As the deployment of Riverbed Steelhead progresses through the fleet, the traffic monitoring capability contained in the Steelheads can be used to help the Navy better identify the amount of SSL/TLS traffic that exists.

2. Diminished Returns Due to Redundant Compression

If no modifications are made to the implementation of Riverbed Steelhead's LZ Compression, the performance achieved by using EXI encoded data is slightly diminished. To be clear, even after the diminished returns on EXI compression, the performance improvement achieved through EXI is still better than using LZ based compression. In order to reap all the benefits of EXI, LZ Compression must be disabled for those traffic flows.

To the author's knowledge, the following technical solutions are not available at this time, but are suggested for further study and possible implementation by Riverbed and other WOC manufactures. The first solution would be to conduct speculative compression. Speculative compression would process network traffic using parallel branches with one branch received compression and the other not. The two branches final size would be compared and only the smallest branch would be transmitted. Speculative compression ultimately requires additional system resources and would inevitably incur additional delays in data transmission. The second solution is to intelligently identify EXI encoded data by performing deep packet inspection on the traffic to locate and recognize the header information present in EXI encoded data. Once EXI encoded data is identified, LZ compression would be suppressed. Ultimately it may be more desirable to incorporate an intelligent method to identify EXI encoded data in the existing Adaptive Compression feature, however it is recognized that the additional processing and latency incurred may make it unjustifiable.

The current Adaptive Compression method was found to not identify previously compressed data, but rather to balance the processing load placed on the central processing unit (CPU) between compression and traffic processing. In order to avoid the application of LZ compression, an in-path rule can be configured for the Riverbed

Steelhead. In-path rules can identify network traffic based on various characteristics such as traffic source and destination addresses. The rule then provides a custom configuration for the associated network traffic, allowing LZ compression to be disabled. This method, however, requires knowledge of the data formats associated with network traffic in order to apply the rule.

The difficulty of identifying network traffic in order to intelligently apply or suppress additional compression highlights a much larger problem faced by the Navy. **The Navy does not have a clear picture of the types of data and the associated amounts being transmitted over their networks.** The lack of a clear network traffic model is potentially a leading cause contributing to the long delay associated with establishing EXI as the single data exchange format for the Navy. Before the Navy can properly apply in-path rules, they must have a clear understanding of what traffic is transiting their networks.

B. ESTABLISHING AN AUDITABLE BANDWIDTH BUDGET

Without the ability to measure the SATCOM resources required by individual network applications and systems, it is difficult to identify where efforts should be focused in order to improve performance. A network traffic model is also necessary in order to make informed decisions with respect to current SATCOM resource allocations and planning for future capacity requirements. With the exception of the scheduled installation of Riverbed Steelhead systems in the fleet, the Navy's method for improving SATCOM performance has been to focus on adding more capacity. As addressed in Chapter II, this approach is very costly and provides less return on investment than compression.

The Satellite Communications Database (SDB) in conjunction with the fleet SATCOM bandwidth requirement message influences the allocation of SATCOM capacity for afloat units. Requirements are determined based on historical capacity allocations and forecasted growth in system and application demands. Accurately capturing SATCOM resource utilization is a dynamic and difficult problem. Applications may only require momentary access to SATCOM resource to transfer their data after

which those resources are released for other systems and applications to use. Therefore it does not make sense to provision SATCOM resources based on capacity requirement defined by individual systems and applications. To improve upon SATCOM resource management, applications should be required to identify the amount of data they generate, how often it is generated, and the time allocated for delivery. Once this baseline is established for each application, a statistical model can be generated representing the amount of network traffic requiring transmission. The network traffic model would also establish the baseline for an auditable bandwidth budget. Identifying critical systems and applications of varying degree and knowing how much data each system and application requires will enable the Navy to better budget for their SATCOM capacity requirements.

Having a defined SATCOM budget will enable the Navy to further refine where SATCOM capacity savings can be obtained. Knowing how much data a system or application is generating is the first step. Identifying what type of data is being generated is the second step. Systems generating structured or semi-structured data that could be formatted as XML would benefit substantially by implementing EXI. The decision to transition to EXI should not be based on the system or applications performance achieved while having full access to allocated SATCOM resources. SATCOM resources can be reduced or removed either by enemy action, inclement weather, equipment failure, or through self-imposed restrictions such as during emission control (EMCON) conditions.

Transitioning candidate systems and applications to use EXI further ensures their availability when operating in a communications contested environment (CCE) where only communication channels with much less capacity are available. In addition, lower capacity channels such as Battle Force Tactical Network (BFTN) and the Mobile User Objective System (MUOS) may not be viewed as feasible options to support mission critical communications when data is formatted in XML. However if efficient data formatting is implemented in systems through EXI, while operating in a capacity rich environment, more of those systems will be supportable in a capacity constrained environment. The time to practice efficient data encoding is not after the high capacity environment is denied, but rather well before.

Developing an accurate network traffic model, auditable bandwidth budget, and identifying candidate systems and applications will take time. Fortunately a foundation already exists through the Baseline Allowance Control (BAC) list established by U.S. Fleet Forces (USFF) and maintained by the fleet functional area manager (Fleet FAM). The BAC list identifies applications and systems authorized for installation on afloat networks by ship class. The BAC list also identifies the status of each system as stand alone, networked, or web app. Expanding the characteristics of each system and application to include the type of data, amount of data, periodicity of transmission, and time required for delivery would establish the beginning of a network traffic model.

C. ESTABLISHING AND ACCELERATING EXI ADOPTION IN THE NAVY

Expanding the BAC list to begin the creation of a network traffic model and follow on auditable bandwidth budget will take time. Fortunately there are actionable steps that can be taken now to incorporate EXI into existing Navy systems and applications. EXI first entered into the DOD through an Air Force Small Business Innovation Research (SBIR) project with AgileDelta (Air Force Research Laboratory, 2003). Navy testing was conducted through JRAE 06 and the Air Force 2006 Joint Expeditionary Force Experiment (JEFX 06) with both tests independently recommending immediate fielding of EXI (Navy SBIR, 2007). Systems used during JRAE 06 and JEFX 06 include GCCS, C2PC, and AFATDS (Schneider, 2008). Those systems and others like them can benefit from the implementation of EXI immediately.

With the advent of the Navy's Consolidated Afloat Networks and Enterprise Services (CANES), applications will utilize shared enterprise services rather than rely upon their own independent hardware and software. This new environment allows for EXI to be implemented as a core enterprise service for all systems and applications to use. The remaining hurdle for the Navy is to identify what command will fund the implementation of EXI services for CANES. Once EXI is made available as an enterprise service, the next step is to use the auditable bandwidth budget to identify systems and applications that may benefit most from EXI and provide requirements and assistance for implementation.

D. FINAL THOUGHTS

This research highlights EXI's complementary role in Navy WAN optimization and presents the necessary steps required to reap the benefits associated with the long overdue implementation of EXI. A network traffic model leading to an auditable bandwidth budget is critical not only to identify candidate systems and applications for future EXI adoption and proper configuration, but also to enable accurate SATCOM capacity allocations and future capacity planning capabilities. The benefits of EXI have been evident for nearly a decade. EXI has existed as a DOD Information Technology Standards Registry (DISR) standard since 2011 (DOD IT Standards Registry, 2011). The appropriate implementation environment now exists through CANES. If the Navy is to remain at the forefront of net-centric warfare it must delay no longer and implement the efficient and compact encoding found in EXI for data transmitted over its networks.

APPENDIX A. DATA SETS AND SCHEMA LOCATIONS

AgileDelta's Chief Technologist, Mr. John Schnieder; provided the following items:

- JRAE and VMF data sets and associated schema
- Refined FSM schema
- TCP tools

Mr. Schneider can be reached at john.schneider@agiledelta.com.

Mr. Matthew Wright from the Navy Supply Systems Command (NAVSUP) Business Systems Center provided the FSM data set. Mr. Wright can be reached at matthew.wright5@navy.mil.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. PROCEEDINGS ARTICLE

While researching this thesis topic, the author collaborated on an article published in the July 2014 issue of the *United States Naval Institute Proceedings* entitled “Being Efficient with Bandwidth” (Debich, Hill, Miller, & Brutzman, 2014). The article is available online at <http://www.usni.org/magazines/proceedings/2014-07/professional-notes>, and is reprinted from *Proceedings* with permission; Copyright © 2014 U.S. Naval Institute (<http://www.usni.org>).

Being Efficient with Bandwidth

By Lieutenant Commander Steve Debich, Lieutenant Bruce Hill, Captain Scot Miller (Retired), U.S. Navy, and Dr. Don Brutzman

Naval information dominance hinges on three fundamental capabilities: assured command and control (C2), battlespace awareness, and integrated fires. None of these are possible without effective communications links. Networks—and more specifically, the information flowing through them—are now a center of gravity for the Fleet.¹ Maritime tactics and operational plans rely on levels of synchronization only possible through high-bandwidth communications. Satellite communication (SATCOM) is the Fleet's primary path for high-bandwidth C2. However, afloat units may be denied access due to equipment failure, technical problems, weather phenomenon, or enemy actions, forcing reliance on lower-bandwidth alternatives.

For afloat units, bandwidth has become a critical but painfully finite resource that must be conserved. SATCOM carries data from a large number of disparate systems often referred to as "stovepipes." These systems vary in function from tactical to administrative, and the data formats for each application vary greatly. The result is communications only occurring vertically within a system, but not across the breadth of different systems. When many such stovepipes contend for access to the same ship-to-shore transport path, even the largest SATCOM channels can become congested. Future assured C2 requires interoperability between stovepipes and better prioritization of network traffic.

Before identifying the solution, we must understand the factors that impose constraints on the transmission path: bandwidth, latency, and throughput.

Bandwidth: Not The Same As "Throughput"

Bandwidth is literally the "width" of the frequency band used to carry a data signal. It is more often described as the transmission capacity of the communications medium, measured in terms of bits per second.² To increase the capacity of an electromagnetic communications channel, modulation technologies and methods would need improvement, or an additional antenna could be installed. Both approaches illustrate significant engineering and financial constraints associated with increasing bandwidth, particularly in the shipboard environment.

SATCOM connections are often depicted as lightning bolts connecting deployed units with relay systems. These lightning bolts convey the impression that data are instantaneously transmitted from unit A to unit B through an optimally placed satellite node. Unfortunately SATCOM transmissions are far from instantaneous: They incur significant delays in comparison to terrestrial communications paths. The combined delay is known as latency.

Latency is an accumulated series of delays that can occur in each step of the communications path between the sender and receiver. Such delays occur as part of propagation delay during signal transmission, network processing and interface delays, varying methods for buffering and queuing, and cumulative router and switch delays.³ Latency from the perspective of network traffic is the delay from the time of the start of packet transmission at the sender host to the time of the end of packet reception at the receiver host.

Unfortunately latency has significant effects on throughput. This is due in part to the degradation experienced by the primary networking protocol TCP when operating over a high latency network.⁴ SATCOM channels routinely operate with latency between 500 to 800 milliseconds. Response "waiting time" is a particular problem for communications protocols like TCP that includes frequent acknowledgement among participants. Increased latency ultimately results in decreased throughput.

Throughput is the rate at which new data—actual information—is transferred through a system. Like bandwidth, it is measured in bits per second and can be considered the actual effective capacity of a channel or the "rate of successful message delivery" being achieved. A common misconception is that bandwidth and throughput are synonymous. Numerous additional constraints can limit the amount of data that can be transferred between two points, such as the overhead of communication protocols and latency delays, which may keep a channel idle. Thus bandwidth indicates the maximum possible data-transfer capacity, while throughput is what capacity actually occurs. Throughput is often significantly lower than the communications channel's bandwidth capacity. Ultimately round-trip time dominates performance more than bandwidth does.⁵

Common Practice: SATCOM

For Navy ships at sea, the only access to high bandwidth is through SATCOM systems. In our increasingly connected world, the value placed on access to high bandwidth continues to rise. As bandwidth increases, the amount of data that can be transferred between two points also increases. As bandwidth is increased, additional capacity is quickly consumed by ever-more sophisticated sensors, unmanned vehicles, and other network-centric dependencies.⁸ Most high-bandwidth paths utilize the super- and extremely-high frequency (SHF/EHF) spectrum for SATCOM communications. Though data and voice circuits exist in other portions of the spectrum, SHF and EHF carry the brunt of Navy traffic, with SHF (C/Ku/X band) ultimately providing the biggest “pipe” for data flows.

In the past, the solution to demand for increasing data transfer was to increase bandwidth, and thereby capacity. As the DoD throttles back spending, many areas must become more efficient in order to accomplish defense missions. Similar approaches for efficiency must be applied with respect to communication systems. The amount of information to be shared is not expected to decrease. Because constraints on SATCOM bandwidth make even marginal increases a costly venture, the Navy must explore new tactics. Perhaps solutions lie not in the channel itself, but in the format of data transmitted. What if we can convey the same information using just a fraction of the original zeros and ones, while at the same time connecting stovepipes through data interoperability?

XML: The Language of Interoperability

Interoperability is essential to the key information dominance capabilities. Shipboard computers must talk to each other, computers from other service branches, and computers from partner nations. To facilitate interoperability, an open-standards approach is critical. The Department of the Navy’s chief information officer has designated the extensible markup language (XML) as the data-definition language of choice for information standardization, and for good reason: It is the *de facto* standard format for systems talking across the web.⁹ By design, XML adds structure to data, which in turn facilitates validation of correctness and system interoperability. XML is the *lingua franca* of the world’s computers.

Though XML is a path to both technical and semantic interoperability, it has an Achilles heel: It was never intended to be compact.⁸ In terrestrial networks with low latency contributing to massive throughput, this is usually unimportant. For the Navy, however, large messages mean slower connections and less information to forward-deployed units relying on SATCOM. Transmitting large messages also draws more power, so XML isn’t ideal for mobile or unmanned devices running on batteries. Viewed in this light, XML is less attractive, but it doesn’t have to be that way. Recent advances in data compression are providing new design options.

Shrinking Data, Broadening the Web

In 2004, the World Wide Web Consortium began to address this issue, and in 2014 released the Efficient XML Interchange (EXI) Format Recommendation.⁹ EXI is an alternate encoding of XML data that leverages the inherent structure of XML to tightly compress it. Since it is designed specifically for XML, the results are superior to generic compression methods. In some cases, EXI compression results in files that are less than 10 percent the size of the original XML file.¹⁰ Perhaps even more surprising is that EXI decompresses faster, using fewer computations and therefore drawing less power than plain text-based ZIP and GZIP compression.

Given that XML enables interoperability, and that EXI shrinks it, Fleet communications architects and program managers should be interested.¹¹ Systems could potentially convert and transmit information in XML format, and with EXI they could send more information in less time. By incorporating EXI, web-based architectures such as CANES and C4I systems using service-oriented architectures may be viable over constrained SATCOM links. Unmanned systems and remote sensors might use EXI to conserve batteries on extended missions. A single file cut to a tenth of its original size is useful in itself, but the aggregate impact over thousands of nodes in a cloud, each sending thousands of files, could be immense.

Other impacts pertain as well. For example, encryption is usually considered independent of compression. However, by randomizing a bit stream, encryption scrambles the structure necessary for effective compression. That means encrypted streams cannot be compressed. Compression must occur before encryption when transmitting, and decompression after decryption on the receiving end. This principle is so important that the order should be checked for all Navy communications channels.

Since message size is just one of many factors in network throughput, EXI is not a silver-bullet for Navy bandwidth woes, but it certainly can’t hurt. It is not mutually exclusive of other attempts to address the issue. Navy communications designers need not choose between a new SATCOM constellation and EXI, or between commercial network accelerators and EXI; they can have both. Considering that EXI is open standard, supports interoperability, and shrinks data the Navy is already sending over its networks, there is little to lose and much to gain. The Navy can be more efficient with a precious afloat resource: bandwidth.

1. Chief of Naval Operations Information Dominance, Navy Strategy for Achieving Information Dominance, 2013-2017, 1 January 2013, www.dtic.mil/docs/citations/ADA571217 [6] .
2. Anu A. Gokhale, *Introduction to Telecommunications* (Cengage Learning, 2004), <http://books.google.com/books?id=QowmxWAOEtYC&pgis=1> [7] , 455.
3. Rony Kay, "Pragmatic Network Latency Engineering Fundamental Facts and Analysis" (cPacket Networks Inc, 2009), http://cpacket.com/wp-content/files_mf/introductiontonetworklatencyengin... [8] .
4. Thomas R. Henderson, Randy H. Katz, *TCP Performance over Satellite Channels* (Berkeley, CA: University of California at Berkeley, 1999), www.eecs.berkeley.edu/Pubs/TechRpts/1999/CSD-99-1083.pdf [9] .
5. Mike Belshe, "More Bandwidth Doesn't Matter (Much)," 2010, www.chromium.org/spdy [10] .
6. Isaac R. Porche, Bradley Wilson, Erin-Elizabeth Johnson, Shane Tierney, Evan Saltzman, RAND Corporation, "Data Flood: Helping the Navy Address the Rising Tide of Sensor Information," 2014, www.rand.org/pubs/research_reports/RR315.html [11] .
7. Department of the Navy Chief Information Officer, "DON policy on the use of extensible markup language (XML)," 2012, <http://xml.coverpages.org/DON-XMLPolicy200212.pdf> [12] .
8. Mike Cokus, Santiago Pericas-Geertsen, "XML binary characterization properties," 2005, www.w3.org/TR/xbc-properties/#xml-design-goals [13] .
9. Takuki Kamiya, Efficient XML Interchange Working Group, 2014, www.w3.org/XML/EXI [14] .
10. Sheldon Snyder, Don McGregor, Don Brutzman, "Efficient XML interchange: Compact, efficient, and standards-based XML for modeling and simulation," 2009, <http://calhoun.nps.edu/public/handle/10945/5422> [15] .
11. Jeffrey Williams, "Document-based message-centric security using XML authentication and encryption for coalition and interagency operations," master's thesis, Naval Postgraduate School, 2009, <http://calhoun.nps.edu/public/handle/10945/4610> [16] .

Lieutenant Commander Debich and Lieutenant Hill are information professional officers studying network operations at the Naval Postgraduate School (NPS).

Captain Miller is a retired information professional officer serving as a research associate at NPS. He is the former commanding officer of the Navy Center for Tactical Systems Interoperability.

Dr. Brutzman is a retired submarine officer working in the information sciences department, Undersea Warfare Academic Group, and MOVES Institute at NPS. Additional insights by Dr. Dan Boger, Captain Louis Unrein (U.S. Navy) and other reviewers are gratefully acknowledged.

Source URL: <http://www.usni.org/magazines/proceedings/2014-07/professional-notes>

APPENDIX C. RIVERBED FLEET INSTALLATION MESSAGE

UNCLASSIFIED/

RTTUZYUW RHOIAAA0004 0482023-UUUU--RHMFIUU.
ZNR UUUUU
R 172023Z FEB 15 ZYB
FM PEO C4I SAN DIEGO CA
TO COMNAVAIRFOR SAN DIEGO CA
COMTENTHFLT
COMTHIRDFLT
COMFIFTHFLT
COMSEVENTHFLT
COMSIXTHFLT
COMUSNAVCENT
COMPACFLT PEARL HARBOR HI
COMNAVAIRLANT NORFOLK VA
COMNAVAIRPAC SAN DIEGO CA
COMNAVSURFPAC SAN DIEGO CA
COMNAVSURFOR SAN DIEGO CA
COMUSFLTFORCOM NORFOLK VA
COMFLTTCYBERCOM FT GEORGE G MEADE MD
PEO C4I SAN DIEGO CA
COMSPAWARSSCOM SAN DIEGO CA
COMSPAWARSSCOM FRD SAN DIEGO CA
CTF 69
CTF 76
COMCARAIRWING ELEVEN
COMCARAIRWING FIVE
COMCARAIRWING FOURTEEN
COMCARAIRWING ONE
COMCARAIRWING SEVEN
COMCARAIRWING SEVENTEEN
COMCARAIRWING THREE
COMCARAIRWING TWO
COMCARSTRKGRU EIGHT
COMCARSTRKGRU NINE
AFLOATRAGRUPACNORWEST EVERETT WA
AFLOATRAGRUWESTPAC YOKOSUKA JA
CENINFODOM CORRY STATION PENSACOLA FL
CENINFODOM UNIT SD SAN DIEGO CA
COMNAVAIRWARCENWPNDIV CHINA LAKE CA
COMNAVSPECWARDEVGRU DAM NECK VA
COMPACFLT DET INTEL READINESS CELL SAN DIEGO CA

IUSS SEA COMPONENT EAST
IUSS SEA COMPONENT WEST
IUSSOPS SUPPCEN LITTLE CREEK VA
MCTSSA CAMP PENDLETON CA
MSFSC NORFOLK VA
NAVCOMTELSTA BAHRAIN
NAVCOMTELSTA NAPLES IT
NAVCOMTELSTA SAN DIEGO CA
NAVIOCOM DENVER CO
NAVIOCOM FT GEORGE G MEADE MD
NAVIOCOM FT GORDON GA
NAVIOCOM HAWAII
NAVIOCOM MEDINA TX
NAVIOCOM SAN DIEGO CA
NAVIOCOM WHIDBEY ISLAND WA
NAVSHIPYD AND IMF PEARL HARBOR HI
NAVSTA INGLESIDE TX
NAVSTKAIRWARCEN FALLON NV
NAVSURFWARCENDIV DAHLGREN VA
NCTAMS PAC HONOLULU HI
NSACSS FT GEORGE G MEADE MD
NSACSS FT GORDON GA
NSACSS HAWAII
ONI WASHINGTON DC
OPNAV DET SITE R FT DETRICK MD
COMCARSTRKGRU ONE
COMCARSTRKGRU THREE
COMCARSTRKGRU FIVE
COMCARSTRKGRU TEN
COMEXPSTRKGRU TWO
COMEXSTRIKGRU THREE
COMEXSTRIKGRU SEVEN
NCTAMS LANT DET HAMPTON ROADS NORFOLK VA
INFO COMMARFORCOM
COMMARFORPAC
COMMARCORSSYS COM QUANTICO VA
COMNAVIDFOR SUFFOLK VA
COMNAVSAFECEN NORFOLK VA
COMOPTEVFOR NORFOLK VA
COMSC WASHINGTON DC
AEGIS TRAREDCEN DAHLGREN VA
AEGIS TECHREP MOORESTOWN NJ
CENINFODOM LS SAN DIEGO CA
CENSURFCOMBATSYS DAHLGREN VA
CENSURFCOMBATSYS DET EAST NORFOLK VA

CENSURFCOMBATSYS DET MAYPORT FL
 CENSURFCOMBATSYS DET NORFOLK VA
 CENSURFCOMBATSYS DET PACNW EVERETT WA
 CENSURFCOMBATSYS DET PEARL HARBOR HI
 CENSURFCOMBATSYS DET SAN DIEGO CA
 CENSURFCOMBATSYS DET WALLOPS ISLAND VA
 CENSURFCOMBATSYS DET WEST SAN DIEGO CA
 CENSURFCOMBATSYS DET YOKOSUKA JA
 CENSURFCOMBATSYSU DAM NECK VA
 CENSURFCOMBATSYSU GREAT LAKES IL
 CG MCCDC QUANTICO VA
 COMAFLOATRAGRU ATLANTIC NORFOLK VA
 COMAFLOATRAGRU NORFOLK VA
 COMAFLOATRAGRUPAC SAN DIEGO CA
 EWTGLANT NORFOLK VA
 FDRMC DET BAHRAIN
 FDRMC DET ROTA SP
 FDRMC NAPLES IT
 IBSSO FT GEORGE G MEADE MD
 MINEWARTRACEN SAN DIEGO CA
 NAVCYBERDEFOPSCOM SUFFOLK VA
 NRO WASHINGTON DC
 MARMC NORFOLK VA
 SOUTHEAST RMC MAYPORT FL
 SOUTHWEST RMC SAN DIEGO CA
 SPAWARSYSCEN ATLANTIC CHARLESTON SC
 SPAWARSYSCEN PACIFIC SAN DIEGO CA
 SPAWARSYFAC PAC YOKOSUKA JA
 SSO NAVY WASHINGTON DC
 NAVNETWARCOM SUFFOLK VA
 BT
 UNCLAS
 COMNAVAIRFOR SAN DIEGO CA/PASS TO SUBORDINATE COMMANDS//
 COMNAVSURFOR SAN DIEGO CA/PASS TO SUBORDINATE COMMANDS//
 COMNAVAIRLANT NORFOLK VA/PASS TO SUBORDINATE COMMANDS//
 COMNAVAIRPAC SAN DIEGO CA/PASS TO SUBORDINATE COMMANDS//
 COMNAVSURFLANT NORFOLK VA/PASS TO SUBORDINATE COMMANDS//
 COMNAVSURFPAC SAN DIEGO CA/PASS TO SUBORDINATE COMMANDS//
 SECINFO/U/-//
 MSGID/GENADMIN,USMTF,2008/PEO C4I SAN DIEGO CA/PMW 160//
 SUBJ/ADNS SERVICE PACK 3 ROLL OUT//
 POC/PATRICK PEMBERTON/CDR/UNIT:PEO C4I SAN DIEGO/-
 /TEL:619.524.7510/SECTEL:510.524.7510
 /EMAIL:PATRICK.PEMBERTON(AT)NAVY.MIL
 /SMAIL:PATRICK.PEMBERTON(AT)NAVY.MIL.SMIL//

GENTEXT/REMARKS/1. IN JUNE 2015, PMW 160 WILL COMMENCE FIELDING THE NEXT GENERATION OF ADNS CAPABILITIES WITH SERVICE PACK 3 (SP3). SP3 INSTALLS WILL BRING IMPROVED WIDE AREA NETWORK (WAN) OPTIMIZATION (WAN-O), CAPACITY INCREASES TO SUPPORT FUTURE FLEET BANDWIDTH REQUIREMENTS, AND SUPPORT FOR PLANNED A2AD CAPABILITIES FROM PEO C4I.

2. SPECIFIC UPGRADES AND CAPABILITIES WILL INCLUDE:

A. WAN-O: SP3 FIELDS RIVERBED WAN OPTIMIZATION, DELIVERING IMPROVED DATA COMPRESSION, QUALITY OF SERVICE, APPLICATION ACCELERATION, AND NETWORK MONITORING TO USERS. EFFECTIVE SATCOM THROUGHPUT WILL BE SIGNIFICANTLY INCREASED, AND USER EXPERIENCE WILL BE IMPROVED.

B. CDLS INTEGRATION: ALLOWS FOR ROUTING OF HIGH BANDWIDTH ISR DATA FROM AIRBORNE PLATFORMS ACROSS NAVY NETWORKS AND ELIMINATES CONNECTIVITY STOVEPIPE ISSUES.

C. EHF THINLINE: SUPPORTS CONNECTIVITY BETWEEN ADNS PLATFORMS VIA EHF TIP WHEN SHORE SERVICES ARE UNAVAILABLE

D. VIPERSAT: SUPPORT FOR SHARING OF AVAILABLE BANDWIDTH BETWEEN CBSP LEASES BY PLATFORMS.

E. VOICE AND VIDEO OVER SECURE IP (VOSIP): FOR FORCE LEVEL INSTALLS, SP3 WILL PROVIDE AN AFLOAT CALL MANAGER ON THE GENSER ENCLAVE.

F. NAVY CONTINUOUS TRAINING ENVIRONMENT: IMPROVES SUPPORT FOR FLEET SYNTHETIC TRAINING BY ALLOWING FORWARD DEPLOYED BMD SHIPS TO PARTICIPATE IN FLEET, COCOM, AND MDA EVENTS WHILE UNDERWAY.

G. PLANNED A2AD IMPROVEMENTS: SP3 SHORE UPGRADES PROVIDE THE FOUNDATION FOR FURTHER A2AD IMPROVEMENTS THROUGH 2020.

H. SHORE CAPACITY INCREASES: ROUTING, CRYPTO, AND WAN-O DEVICES AT THE SHORE WILL INCREASE UNIT BANDWIDTH CAPACITY AND TOTAL UNITS SUPPORTED. THIS WILL IMPROVE SUPPORT IN AOR'S NEARING MAXIMUM THROUGHPUT TODAY, AND PROVIDE SUFFICIENT BANDWIDTH FOR PLANNED ISR DATA FROM AIRBORNE PLATFORMS JOINING THE ADNS WAN.

3. ASHORE IMPACTS: DURING SHORE INSTALL PERIODS, DISRUPTIONS OF SERVICE MAY OCCUR. PMW 160 WILL WORK WITH THE AFFECTED SHORE SITE, NAVIDFOR, AND FLT STAKEHOLDERS TO OBTAIN REQUIRED APPROVALS AND KEEP ALCON INFORMED. ADNS IS DEVELOPING A SERVICE BY SERVICE CUTOVER PLAN TO MINIMIZE SPECIFIC OUTAGES OF SERVICES TO MINUTES AS NEW SP3 DEVICES ARE BROUGHT ONLINE.

4. AFLOAT IMPACTS: SP3 WILL BE FIELDIED TO AFLOAT UNITS PLANNED FOR ADNS INSTALLS BEGINNING IN JUN15. SP3 SHIPS WILL BE ABLE TO CONNECT TO NON-SP3 SHORE SITES WHILE THE SHORE ROLL OUT IS IN PROGRESS. SP3 SHIPS WILL ONLY HAVE FULL WAN-O CAPABILITY WHEN CONNECTED TO AN SP3 SHORE SITE.

5. PLANNED SHORE INSTALL SCHEDULE:

JUL 15 NCTAMS PAC

FEB 16 NCTS NAPLES

MAY 16 NCTS BAHRAIN

AUG 16 NCTAMS LANT

6. THE PROGRAM OFFICE FOR TACTICAL NETWORKS LOOKS FORWARD TO DELIVERING THESE ADVANCED CAPABILITIES TO THE WARFIGHTER SOON. FOR FURTHER DETAILS ON SP3, PLEASE CONTACT THE FOLLOWING:

CDR PATRICK PEMBERTON, ASSISTANT PROGRAM MANAGER FOR ADNS

PATRICK.PEMBERTON(AT)NAVY.MIL COMM: 619-524-7510 DSN: 510-524-7510

LISA SALISBURY, ADNS LEAD ENGINEER

LISA.SALISBURY(AT)NAVY.MIL COMM: 619-524-3567; DSN: 510-524-3567//

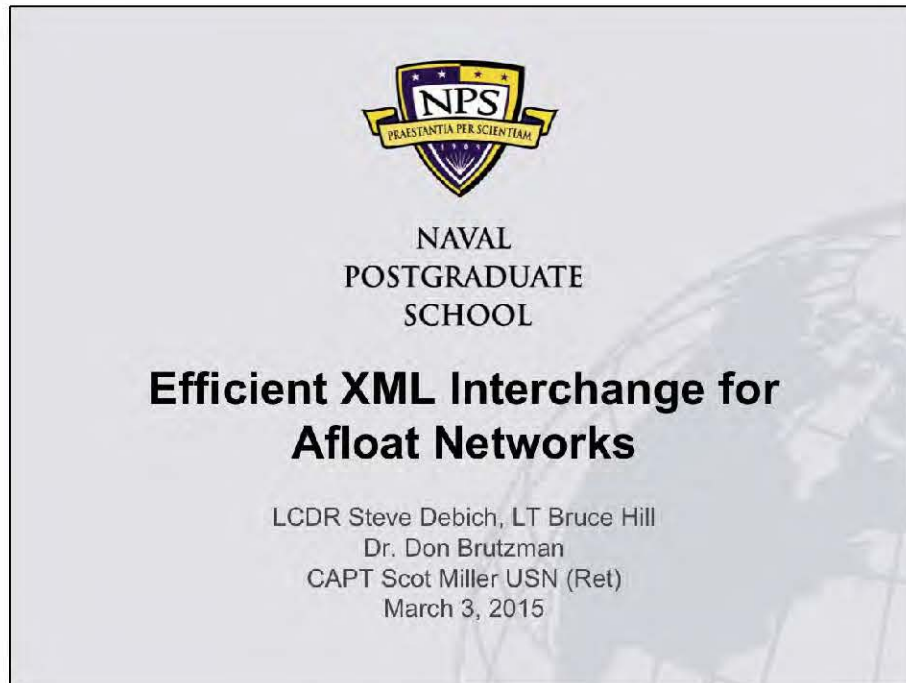
BT


#0004

NNNN

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. PRESENTATION PREPARED FOR NAVIDFOR



NAVAL
POSTGRADUATE
SCHOOL

BLUF

- Capacity alone **does not** improve the Navy's SATCOM network performance.
- **Compression** improves performance of existing bandwidth for significantly less cost.
- **We can improve** network performance for structured data (tactical, sensor, administrative) **by at least 50%.**
- By combining network-based compression and host-based compression, **even greater performance improvement is possible.**

2

Bullet 1:

Bold statement to make a point! Accurate but oversimplified statement pertaining to the multivariable problem of SATCOM network performance.

Bullet 2:

Less expensive to compress data than it is to launch another satellite, or develop more aggressive encoding and modulation schemes.

Bullet 3:

Fleet trials show Riverbed to provide a bandwidth capacity increase of .5; therefore an 8 Mbps channel performs like a 12 Mbps channel.

Bullet 4:

Combining EXI with Riverbed, the lowest improvement is

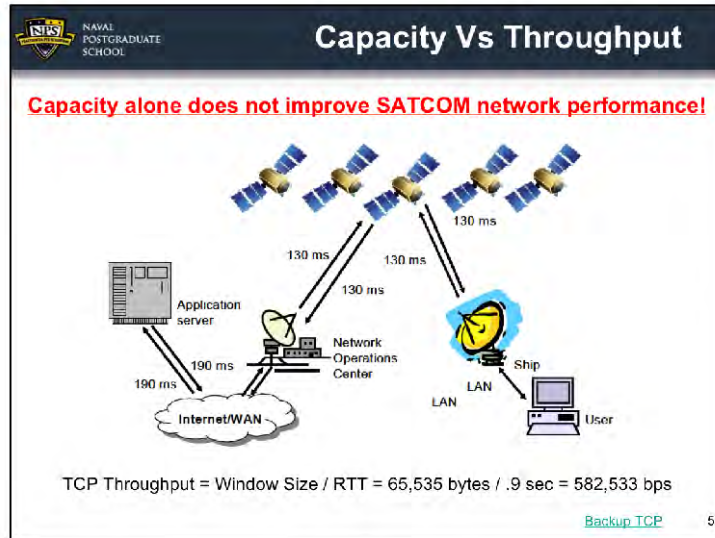


- The Role of EXI in WAN Optimization
 - SATCOM performance limitations
 - WAN Optimization Techniques
 - WAN Optimization in the Navy
 - XML/EXI and Navy Comms
 - EXI and the DoD
 - Combining WAN Optimization and EXI
 - Testing and Results
 - The Way Forward
- Evaluating EXI
 - Small Files: XML, JSON and their binary encodings
 - Large Files: Extending EXI evaluations
 - Best practices and conclusions



LCDR Steve Debich

THE ROLE OF EXI IN WAN OPTIMIZATION



Adding more capacity alone will allow more individual TCP connections to exist, but each connection will not achieve greater than 582,533 bps.

Capacity vs Bandwidth: can discuss as needed; Bandwidth is the common term used to describe the data carrying capacity of a transmission link. Bandwidth is also synonymous with the range of frequencies (width) in a certain band of the electromagnetic spectrum; i.e. Band-Width. The capacity of a link can vary depending on other variables even if the frequencies in use remain the same. For the remainder of this brief, capacity will be the term used.

Graphic adapted from:

Bentrup, J., Otte, E., Chan, B., Vavrichek, D., & Gingras, D. (2012). *At-sea testing of a wide area network optimization device*. Center for Naval Analysis (CNA).


** add details about lab

Bentrup et al. use 64 KB for window size, arriving at a throughput of 569 Kbps.

Window size is determined by the 16 bit "Window" field in the TCP header. This results in a window size of $2^{16} = 65,535$ bytes. Using the more exact number for Window size results in the max throughput = 582,533 bps.

From Bentrup et al.


The large latency associated with satellite communications is due to the amount of time required for data packets to travel at the speed of light from the earth's surface to a geo-



NAVAL
POSTGRADUATE
SCHOOL

WAN Optimization: Eating Elephants

- Long Fat Network (LFN): pronounced "Elephant"
- Bridging the LAN/WAN disparity requires optimization
- Requires maximum efficiency
 - WAN optimization + EXI



6

Long = length as measured in time latency (one way), round trip (two way)

Fat = capacity

The capacity (bandwidth) X round trip (delay) = bandwidth delay product.

A network with a bandwidth delay product greater than 100,000 bits is known as a long fat network (LFN), with long referring to the length and fat referring to the capacity (RFC 1027, Jacobson).

“How do you eat an elephant?: One bite at a time!”

Title of popular business management book by Bill Hogan

In order to bridge the LAN/WAN disparity, optimization and acceleration must be used.

Next slide comments discuss the other protocols accelerated by WAN optimization.

Below is the same information covered on the back up slides for Capacity, Throughput, and BDP:

<http://www.netcraftsmen.com/tcp-performance-and-the-mathis-equation/>

For those not familiar with it, the buffering required in a receiving system for maximum performance is based on the BW-Delay Product (BDP). It is the amount of data that can be sent between ACKs. You multiply the path's minimum bandwidth by the round-trip delay. The table below shows the buffering requirements for

- Protocol Optimization
- Caching
- Prefetching
- Compression & Data Deduplication
 - Compression: implements algorithms to analyze and consolidate a predefined group of data
 - Data Deduplication: specialized form of compression; also called data suppression
 - Can also improve Protocol Optimization, Caching, and Prefetching causing a cascading beneficial effect

Protocol Optimization:

There is a laundry list of protocols that can be optimized. The disparity of the LAN and WAN environment often results in protocols being designed for LAN characteristics which then has detrimental effects when implemented over a WAN. An application may run just fine when your server and workstations are all on the same 100 Mbps Ethernet LAN, but when you stretch business out across the globe and that application has to traverse many different networks to arrive at the server, chances are it won't perform the same.

Common protocols to optimize:

TCP: Transmission Control Protocol; used to reliably transmit the majority of network traffic.

CIFS: Common Internet File System, also called Server Message Block (SMB). Mainly used for shared access to files, printers, port, etc.

HTTP: Hypertext Transfer Protocol, common protocol for web traffic.

FTP: File Transfer Protocol

NFS: Network File System; similar to CIFS, but NFS is stateless in nature whereas CIFS is extremely stateful.

MAPI: Messaging Application Programming Interface is a messaging architecture and a Component Object Model based API for MS Windows. Allows client programs to become (e-mail) messaging-enabled, -aware, or -based by calling MAPI subsystem routines that interface with certain messaging services. Commonly used with Remote Procedure Call (RPC), the protocol that MS Outlook uses to communicate with MS Exchange.



WAN Optimization and the Navy

- ADNS architecture
- Blue Coat PacketShaper
- Defense Acquisition Challenge funded the testing of WOC vendors
- Riverbed Steelhead testing
 - Trident Warrior 2010 - CNA & SPAWAR on BHR
 - 2012 on board NIM
- Riverbed Steelhead provided 40% more throughput than PacketShaper
- Fleet implementation July 2015
 - (MSG DTG R 172023Z FEB 15)

8

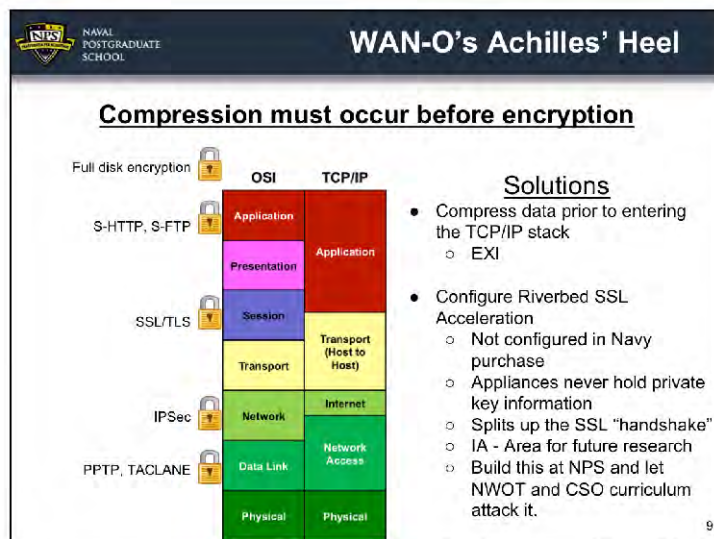
- WAN Optimization is a function performed within the ADNS architecture.
- Blue Coat PacketShaper - Current Navy WAN Optimization Controller (WOC)
- Defense Acquisition Challenge was used to fund the testing of WOC vendors. Submitted by Eric Otte
- Trident Warrior 2010 - CNA & SPAWAR test Riverbed Steelhead on BHR; 2012 NIM Sea Trials.
- Steelhead provided 40% more throughput than PacketShaper.
- Steelhead is officially scheduled for Fleet implementation. (MSG DTG R 172023Z FEB 15)

Shore installations start in July 2015

Details of PacketShaper and Steelhead history provided by Eric Otte, and CDR Patrick Pemberton APM of PMW 160.

PacketShaper's focus was on providing network traffic visibility. Compression and Acceleration were "add ons".

My fleet experience indicates that PacketShaper's compression and acceleration were unstable at best. We had to routinely turn off compression simply to be able to browse the



Encryption seeks to make data highly random; Compression seeks to remove redundancy from data; Therefore Compression must occur before encryption.

Without Riverbed SSL acceleration, the amount of data that can be compressed is limited. You will see why this is a limiting factor when I present my test results.

Larger than necessary network traffic is found in SSL/TLS traffic because compression was removed due to interactions between the protocol's encryption implementation and the application of compression.

Compression Ratio Info-Leakage Made Easy (CRIME) exploits the side-channel vulnerability enabling a skilled attacker to decrypt traffic, enabling cookie stealing and session take-over. Due to the security vulnerability exploited by CRIME, browsers have removed the option for compression over TLS connections (Clark & Van Oorschot, 2013, p. 513)


Riverbed SSL Acceleration

The information below is taken from the Riverbed Feature Brief: SSL Acceleration.

The security issues arise in the area that goes by the name of "key management:" the weak link in an encryption system like SSL is typically protecting the private key information.

The more times you have to handle a key, and the more places you have to store the key, the higher the risk of an inadvertent disclosure of the key.

This challenge of key management explains why it's not sensible to make a symmetric

NAVAL
POSTGRADUATE
SCHOOL

XML/EXI and Navy Comms

- Net-centric warfare, cloud computing, distributed sensor nets require
 - 1. Interoperable data
 - 2. High network throughput
- Extensible Markup Language (XML) is interoperable, but verbose
- Efficient XML Interchange (EXI) is a compact encoding and compression standard for data that retains the interoperable characteristics of XML
- XML interoperability is now viable for afloat networks

10

John Schnieder:

To fight using the web as a weapon system we must have a single standard for all data transfer.

EXI provides that standard.

With a SINGLE standard, data exchange between systems become a non issue

Evolution of Compression	
Generic Compression	XML Compression
<ul style="list-style-type: none"> • 1952 Huffman coding • 1977 Lempel and Ziv developed a universal compression system • 1984 Terry Welch develops LZW from LZ78 • 1994 Burrows-Wheeler transform • 1996 BZip2 incorporates Burrows-Wheeler transform and Huffman coding • 2001 LZMA incorporates LZ and Markov Chaining 	<ul style="list-style-type: none"> • 1998 XML becomes W3C Recommendation • 2000-2005 Multiple XML-specific compression methods emerge. W3C working group formed • July 2007 EXI public working draft • 10 March 2011 EXI becomes W3C Recommendation

Wikipedia:

http://en.wikipedia.org/wiki/Timeline_of_algorithms

1952 - [Huffman coding](#) developed by [David A. Huffman](#)

1977 - [LZ77](#) algorithm developed by [Abraham Lempel](#) and [Jacob Ziv](#)

1984 - [LZW](#) algorithm developed from [LZ78](#) by [Terry Welch](#)

1994 - [Burrows-Wheeler transform](#) developed by [Michael Burrows](#) and [David Wheeler](#)

2001 - [Lempel-Ziv-Markov chain algorithm](#) for compression developed by [Igor Pavlov](#)

From Kattan:

Lempel and Ziv developed a universal compression system in 1977 known as LZ77

A well-known composite compression system is Bzip2.

Bzip2 is a free, open-source, composite and lossless compression algorithm developed in 1996 [9].


Bzip2 compresses files using the Burrows-Wheeler transform and the Huffman coding.

Another composite compression system is the Lempel-Ziv

Markov-chain Algorithm (LZMA) [2]. LZMA uses a similar approach to Deflate. The difference is that it uses range encoding instead of Huffman coding (the range encoder is an integer-based version of Arithmetic Coding) [2]

[2] David Salomon, Data Compression: The Complete Reference, Second edition, 2004.

EXI:



NAVAL
POSTGRADUATE
SCHOOL

Cost Comparison

- WGS = 10 X DSCS
 - \$353.9 million per satellite
 - Total life of program (LoP) cost \$3.73 billion
 - Remember: **“capacity alone does not improve performance”**
- Riverbed provides 50% improvement in throughput
 - Costs are taken out of ADNS sustainment
 - \$2 million - includes installation of ship and shore equipment
- EXI
 - Implement as an Enterprise Service
 - \$100s of thousands or low millions

12

WGS costs retrieved from:

<http://www.bga-aeroweb.com/Defense/Wideband-Global-SATCOM.html>

Supported by:


<http://www.bga-aeroweb.com/Defense/Budget-Data/FY2015/WGS-USAF-PROC-FY2015.pdf>

Riverbed ballpark costs provided through personal communication with ADNS APM CDR Patrick Pemberton

Exact number is unknown, but I have high confidence that estimates in the low single digit millions are accurate

EXI ballpark costs derived from estimated figures presented by AgileDelta's Chief Technology Officer, John Schneider.

Exact number cannot be known at this time without a better plan for implementation. I have high confidence that estimates in the upper hundreds of thousands to low single digit millions are accurate. The bottom line is that it costs a lot less than additional satellites and does not require any new hardware.



NAVAL
POSTGRADUATE
SCHOOL

EXI and the DoD

- July 2003 Air Force SBIR/STTR (AF03-094) - Phase I
- August 2004 - September 2007 (AF03-094) - Phase II
- 2005 - XMPP Chat becomes DISR Mandated Standard
- 2006 - JRAE '06 / JEFX '06
- July 2007 – Feb 2011 Navy SBIR (AF03-094 carry over)
- 2007 EXI submitted to DISR as a "New Emerging Standard" - DISR does not list draft standards; not added
- October 2011 - EXI added to DISR as a standard (submitted by USN)
- December 2012 - EXI becomes a DISR Mandated Standard for DoD
- 2014 - EXI for XMPP Standard in final review status

ASAP: Implement EXI as an Enterprise Service


13

Started as an Air Force SBIR/STTR
 Small Business Innovation Research / Small Business Technology Transfer

AF SBIR number: AF03-094
 SBIR Title:
 Efficient XML: Taking Net-Centricity to the Edge
 Company: AgileDelta

Phase 1: Started July 2003
 Proposal Title: Extending the Infosphere to Mobile Platforms Using Optimized COTS Technologies
 Phase 2: Started August 2004 end September 2007

Abstract:
 The objective of this proposal is to demonstrate methods for extending the Battlespace Infosphere to mobile platforms using optimized COTS technologies. The 1999 USAF Scientific Advisory Board report titled Building the Joint Battlespace Infosphere identifies the eXtensible Markup Language (XML) as one of the most promising technologies for representing JBI information objects. Indeed,

NAVAL POSTGRADUATE SCHOOL

Combining Riverbed and EXI

- Riverbed Steelhead: Network-based Optimization
 - LZ-based compression
 - Scaleable Data Reduction (SDR) (Data Deduplication)
- EXI: Host-based Optimization
- EXI and Riverbed Steelhead are complementary
 - Total Riverbed Compression = LZ + SDR
 - Combined capability = EXI + SDR

14


It is not a competition between Riverbed Steelhead and EXI, but rather a complementary relationship.

Both are good; together they are amazing.

EXI combines advanced encoding and compression

Combining Riverbed with EXI provides additional compression of XML and structured data through the use of EXI while retaining the significant deduplication capability provided by SDR.


Keep in mind that the data deduplication benefits are only applicable to unencrypted data or require the Riverbed SSL acceleration to be configured.



NAVAL
POSTGRADUATE
SCHOOL

What is SDR

- Riverbed proprietary data deduplication process
 - “Cold Transfer” = first transmission
 - Small overhead incurred = “Price of Admission”
 - “Warm Transfer” = second transmission
 - Exploits redundant traffic flows across applications
 - Benefits easily justify the “Price of Admission”
 - Data must be visible to the WOC
 - Encrypted data is not visible



15

Data deduplication is not proprietary, but the exact way that Riverbed implements it to achieve speed and performance is patented. Coordination continues with Riverbed to review details of SDR operations.

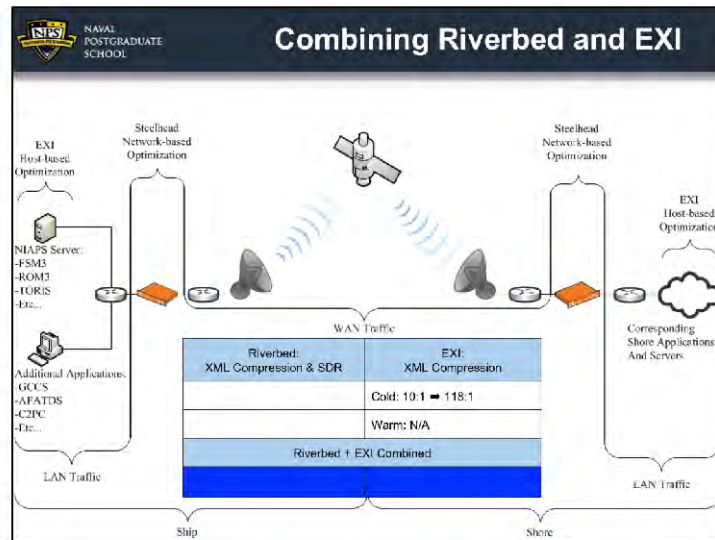
[Eric Otte]

... is very effective at exploiting and compressing redundant traffic flows. Examples include, two or more users going to the same website, an email attachment being sent to multiple participants, message or track traffic where much of the content is repeated/redundant. These are the big wins we saw in our Fleet trials.

-Message traffic coming to a ship often has many of the same PLADs listed for the group. This could only be transmitted once, and then followed by an SDR reference.

There is a price of admission associated with applying SDR. As the data segments are generated and references assigned, the references must be sent across the WAN to the peer WOC so that they two systems share the library. This reference transfer adds overhead which decreases the overall performance of compression previously applied by increasing the total payload sent.

[Figure 3] is an example of referencing data segments and building them into larger segments. In this example, a file (or portion of a file) is initially split into nine segments. As the system learns the data patterns, it combines segments 1 and 2 to form segment 10, and segments 3 and 4 are combined into segment 11. With additional pattern learning, segments



This slide sets up the transition to the testing and results. The numbers below will be filled in on slide 24.

Riverbed Steelhead XML Cold Compression

Cold: 2:1 ➔ 16:1

EXI Compression

10:1 ➔ 118:1


At a minimum EXI compression achieves a 1.61 times greater performance improvement over SH Comp and at most a 19.38 times greater performance improvement over SH Comp.

Riverbed Steelhead XML Warm

Warm: 14:1 ➔ 1,238:1

EXI + Warm: 596:1 ➔ 7,165:1

At a minimum EXI SDR warm transfers achieve 1.32 times greater performance improvement over XML SDR warm transfers and at most 19.05 times greater performance improvement over XML SDR warm transfers.



NAVAL
POSTGRADUATE
SCHOOL

Selecting Representative Datasets

- Food Service Management 3.0 (FSM) database export
 - Total of 94 XML files each representing a database table
 - Four files were selected for testing
- JRAE '06 Target Data
 - Total of 911 targets
 - Separated into batches of 1, 10, 50, and 100 targets
- MIL-STD-6017 Variable Message Format (VMF) Sensor and Track Data
 - Total of 55 sensor and 64 track messages
 - Subset of 2 sensor, and 1 target message selected for testing

17

Administrative, known previously tested mission data (used in JRAE), sensors
 One size does not fit all; results show a range of performance possibilities.
 FSM

-Original data set was to come from Retail Operations Management (ROM 3). The program was in the process of a fleet wide upgrade so finding anyone who had time to generate a data set for me was difficult.

-Food Service Management was suggested to me by NAVSUP since it used the same NIAPS server.

Navy Information Application Product Suite (NIAPS)

Test data consisted of an FSM SQL database exported in XML format. The entire export consisted of 94 individual XML files of various lengths. Of the 94 files, four files were selected for testing, with the first file being the smallest, the last file the largest, and the remaining two files falling in between.

MCM-9_000_Header.xml 648

MCM-9_001_Branch_Personnel_Type.xml 26,701

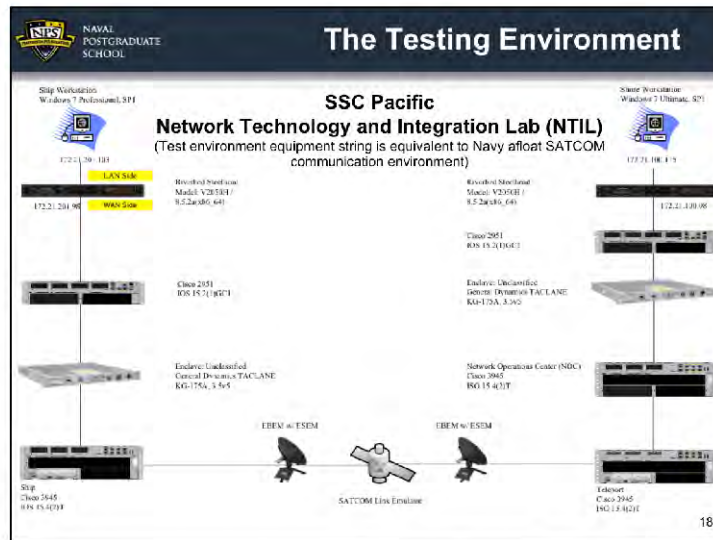
MCM-9_050_Menu_Meal_Recipe_Assoc.xml 68,786,829

MCM-9_084_Planning_Transaction_Dtl.xml 291,223,416

JRAE


911 different targets sent as SOAP messages by AFATDS as part of Joint Target Management (JTM)

Cursor on Target (CoT) data. In XML format.



Equipment string used at the SSC PAC Network Technology and Integration Lab (NTIL)
 POC for lab:
 Eric Otte
eric.otte@navy.mil

This is one lab used to test fleet configurations. Mainly focuses on future implementations.
 Additional ADNS labs have exact equipment string replicas for some ships.



NAVAL
POSTGRADUATE
SCHOOL

Research Questions


- How can host-based and network-based optimization be paired together to produce the best results?
- Tests conducted
 - Compare Steelhead LZ and EXI compression
 - Impact of adaptive compression
 - Effect of Steelhead compression + SDR on cold transfer
 - Effect of Steelhead compression + SDR on warm transfer
 - Effects of a single modification at the beginning, middle, and end of a file
 - Effect with more than one modification per file

19

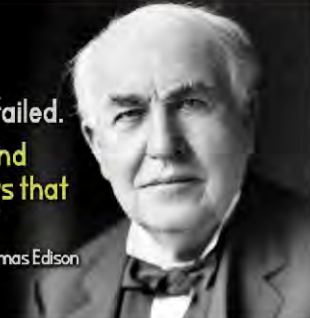
We know Riverbed is coming to the fleet
 We know EXI provides greater compression for XML structured data.
 How can we best combine these two technologies to get the greatest benefits?

In support of the primary research question I ran the tests listed here to identify which features and characteristics contributed most to the overall performance.

1. Can EXI provide better compression than Riverbed Steelhead's LZ compression for XML data?
2. Does Riverbed Steelhead LZ compression performance vary with compression level?
3. What effect does Riverbed Steelhead Adaptive Compression have on Gzip and EXI compressed data?
4. What effect does Riverbed Steelhead Compression + SDR have on XML data and EXI compressed data when transmitted as a cold transfer?
5. What effect does Riverbed Steelhead Compression + SDR have on XML data and EXI compressed data when transmitted as a warm transfer?
6. What effect does Riverbed Steelhead SDR have on XML data and EXI compressed data when a single modification is made at the beginning, middle, and end of a file?
7. What effect does Riverbed Steelhead Compression + SDR have on XML data and EXI compressed data when a single modification is made at the beginning, middle, and end of a file?
8. What effect does Riverbed Steelhead SDR have on XML data and EXI compressed data

NAVAL
POSTGRADUATE
SCHOOL

Travel, Testing & Lessons Learned



"I have not failed.
I've just found
10,000 ways that
won't work."
~ Thomas Edison

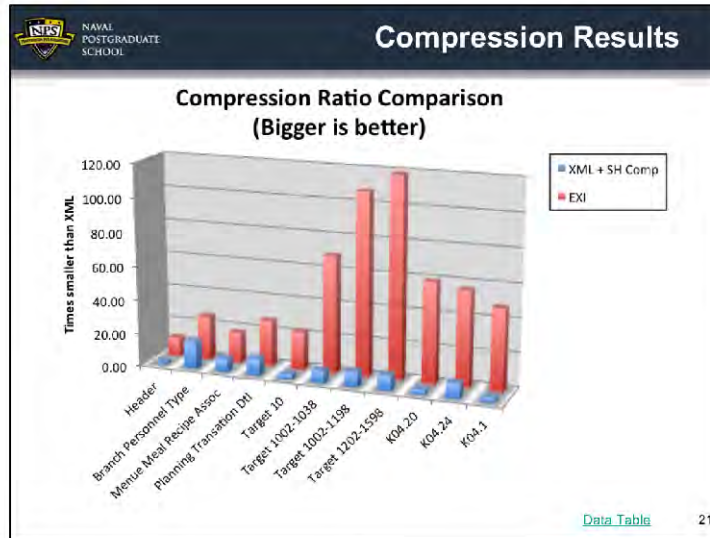
Person Excellence.co

- Three rounds of testing required + remote lab support.
- Thank you OPNAV Studies Program for travel support.

[Lessons Learned details](#) 20

Sincere appreciation goes out to Eric Otte and Britney Chan at the Network Technology and Integration Lab for their continuous support of my testing efforts. Special thanks goes out to Britney for conducting testing for me via e-mail in order for me to refine my testing data sets and methods.

Additional special thanks goes to John Schneider at AgileDelta for his near real-time responses to my e-mails and never ending questions. Without John's keen eye and ability to quickly identify issues with my results data, this testing would not have been completed.



The following graphs depict compression ratios.
 The bigger the compression ratio, the better the performance.
 EXI is better in each test case; there are no test cases where EXI is not better.

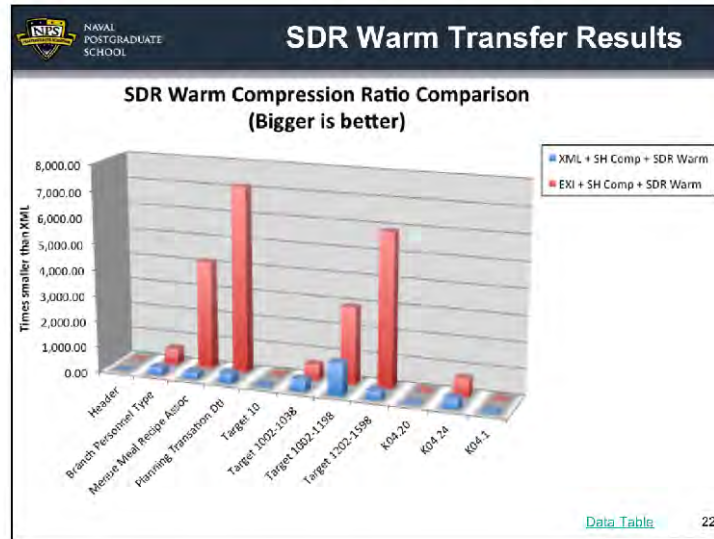
Riverbed Steelhead XML Cold Compression

Cold: 2:1 ➔ 16:1

EXI Compression

10:1 ➔ 118:1

At a minimum EXI compression achieves a 1.61 times greater performance improvement over SH Comp and at most a 19.38 times greater performance improvement over SH Comp.



The following graphs depict compression ratios.
The bigger the compression ratio, the better the performance.

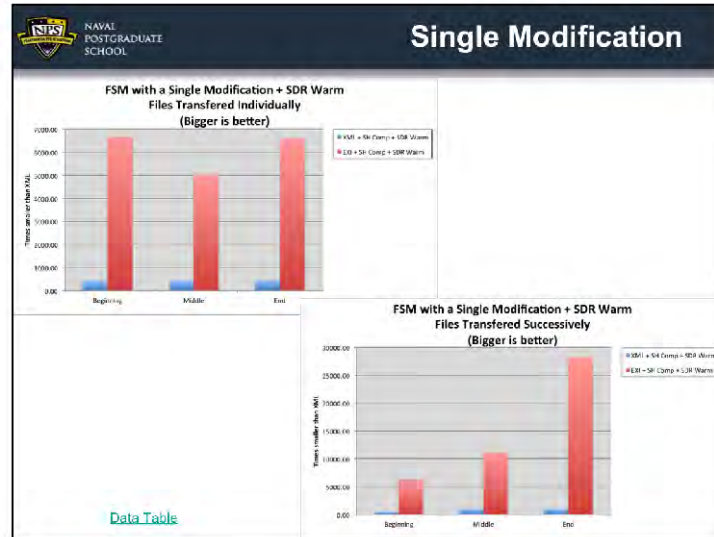
Warm XML Ratios: 14.7:1 to 1,238:1

Warm EXI Ratios: 596:1 to 7,165:1

At a minimum EXI SDR warm transfers achieve 1.32 times greater performance improvement over XML SDR warm transfers and at most 19.05 times greater performance improvement over XML SDR warm transfers.

The 4 test cases that did not have SDR applied to them only receive the benefits seen in Cold compression.

Warm transfers more than make up for any overhead incurred by the cold transfer or “Price of Admission”



The following graphs depict compression ratios.
The bigger the compression ratio, the better the performance.

Individually

Warm XML Ratios: 445:1 to 471:1

Warm EXI Ratios: 5,015:1 to 6,662:1

At a minimum EXI SDR warm transfers achieve 11 times greater performance improvement over XML SDR warm transfers and at most 15 times greater performance improvement over XML SDR warm transfers containing a modification made at the beginning, middle, or end when transferred individually..

-represents sending nearly identical files; two users

Successively

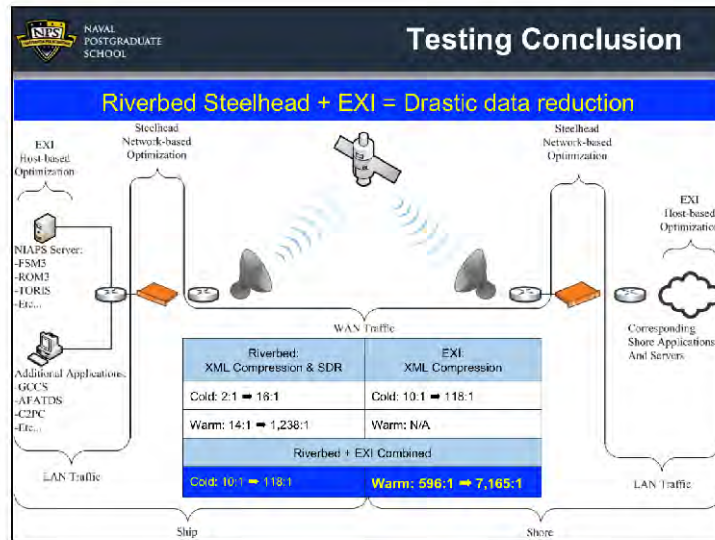
Warm XML Ratios: 466:1 to 1,045:1

Warm EXI Ratios: 6,377:1 to 28,228:1

At a minimum EXI SDR warm transfers achieve 11 times greater performance improvement over XML SDR warm transfers and at most 27 times greater performance improvement over XML SDR warm transfers containing a modification made at the beginning, middle, or end when transferred successively.

-represents sending nearly identical files; three users.

The next slide will hopefully look familiar. Only now I have results to fill in the blanks.



Riverbed Steelhead XML Cold Compression

Cold: 2:1 ⇒ 16:1

EXI Compression

10:1 ⇒ 118:1


At a minimum EXI compression achieves a 1.61 times greater performance improvement over SH Comp and at most a 19.38 times greater performance improvement over SH Comp.

Riverbed Steelhead XML Warm

Warm: 14:1 ⇒ 1,238:1

EXI + Warm: 596:1 ⇒ 7,165:1

At a minimum EXI SDR warm transfers achieve 1.32 times greater performance improvement over XML SDR warm transfers and at most 19.05 times greater performance improvement over XML SDR warm transfers.



NAVAL
POSTGRADUATE
SCHOOL

Testing Summary

- Results vary depending on source of data
- EXI Compression: 1.61 - 19.38 times greater performance
- Adaptive Compression does not avoid additional overhead
- SDR Cold: EXI incurs less decrease in performance than XML
- EXI + SDR Warm: 1.32 - 19.05 times greater performance
- EXI + SDR Warm Single modification:
 - 11-15 times greater performance (individually)
 - 11-27 times greater performance (successively)
- EXI + SDR Warm with more than one modification:
 - Ranges between results achieved by SDR Cold to SDR Warm
 - Depends on amount of commonality between files

26

Mileage may vary

Compression:

At a minimum EXI compression achieves a 1.61 times greater performance improvement over SH Comp and at most a 19.38 times greater performance improvement over SH Comp.

Adaptive compression does not avoid adding additional overhead to pay load.

Possible area of research is how can traffic be processed in parallel (with and without compression) and avoid sending any data that gets larger

SDR Cold:

EXI data incurs less of a performance decrease than corresponding XML data

SDR Warm:


At a minimum EXI SDR warm transfers achieve 1.32 times greater performance improvement over XML SDR warm transfers and at most 19.05 times greater performance improvement over XML SDR warm transfers.

Easily justifies incurring the "Price of Admission" for a cold transfer.

SDR Warm Single Modification:

11-15 sent individually

11-27 sent successively



NAVAL
POSTGRADUATE
SCHOOL

The Way Forward

- Implement EXI as an Enterprise Service through CANES for applications to utilize.
- Develop an auditable bandwidth budget
 - How much am I spending and where?
 - BAC list as a starting point
- Develop metrics to assess Navy applications for EXI eligibility/candidacy
 - Net-centric systems, tactical COP, chat, messaging, etc.
 - N/A: Video, graphics, audio, etc.
 - Identify traffic affected by SSL/TLS or other encryption
- Conduct fleet network traffic analysis - establish baseline
- Prioritize heavy traffic for earliest EXI implementation

26

There is no one-size-fits-all solution.

EXI needs to be an Enterprise Service. The service should be provided by CANES so that any application that can make use of EXI does not need to resource it.

Individual applications would then be responsible for schema development.

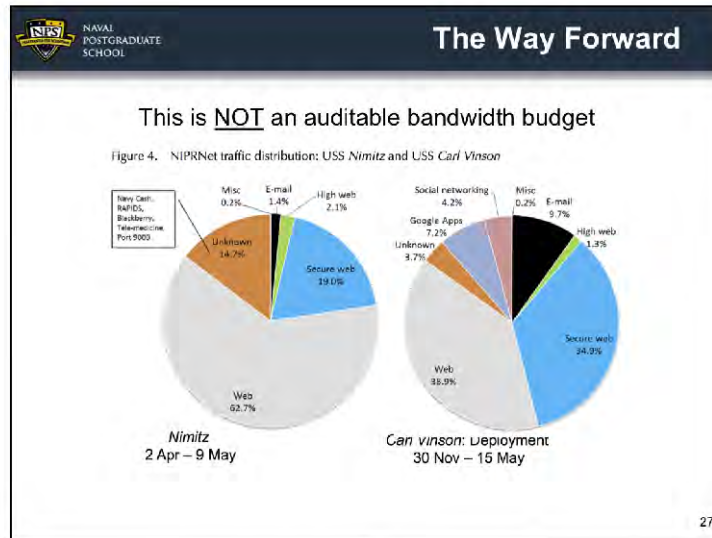
The first step is to be able to know what you have. This requires an auditable bandwidth budget. I can't manage my money if I don't know where I'm spending it.

All applications on the Baseline Allowance Control (BAC) list should be expanded to include the method they use to transmit data, the average amount transferred over a given time, and the max amount transferred at any one time.

Create a working group to identify the criteria and characteristics that would make an applications data a candidate for EXI

Description from John Schneider:

"Regarding how to best implement EXI for the Navy, this is not an insignificant topic to tackle. The Navy is a large and diverse enterprise and there's not a one-size-fits all strategy for deploying any new technology. There are many different deployment scenarios and the Efficient XML Gateway is just one of several different strategies. E.g., if we're talking about net-centric systems, which use web-service protocols, the Efficient Web Services plug-ins are generally the best solution. This is what we used to integrate EXI into GCCS-M, AFATDS and C2PC for JRAE. If we're talking about TCP or UDP applications, TCP/UDP proxies or native integration via XML APIs might be best. If we're talking about aircraft,



The traffic distribution pie-graph is not an auditable bandwidth budget.

It is a start to know where your bandwidth is being used in a snapshot in time, but does not have a high enough resolution to identify the specific application or set of applications that should be targets first and by what methods (general compression, format specific compression)

I envision more of an excel spreadsheet like the BAC list combined with the NIAPS Application Tracker.

Tracker would identify: program name, version, BAC, DADMS, POC, Agency, Data Formats, Type of Transfer Method, Protocols, Amount of data transferred, Replication use?
DADMS = DON Application and Database Management Systems
BAC; US Fleet Forces Functional Area Manager (FAM) Baseline Allowance Control (BAC).

Make an initial assessment of which apps are the heavy BW users. Start with them.


Figure 4 is taken from:

At Sea Testing of Wide Area Network Optimization

Nimitz 2012

Written by:

John Bentrup • Eric Otte (SPAWAR) • Britney Chan (SPAWAR) • Diane Vavrichek • Don



**NAVAL
POSTGRADUATE
SCHOOL**

The Way Forward

Name of application(s)	Application/Database name	BAC Identification Number	Version number	Current Description Date, AIA, LARMS	Data entry last last modified	If replication is listed on another system, indicate the system name	If the app is not listed (BAC), indicate the status, or a link to the source of data	Overall summary of findings (the questionnaire)
Application/Database Name	System Name	BAC #	Version	Description Date, AIA, LARMS	Data entry last last modified	Replication System	Comments	Overall summary of findings (the questionnaire)
Admission Tracking System	ATM	1234	1.0	1/1/2004	1/1/2004	ATM (ATM)	ATM is a web-based application designed to track admission status and provide a link to the source of data.	ATM is a web-based application designed to track admission status and provide a link to the source of data.
Admission Tracking System (Database)	ATM	1234	1.0	1/1/2004	1/1/2004	ATM (ATM)	ATM is a web-based application designed to track admission status and provide a link to the source of data.	ATM is a web-based application designed to track admission status and provide a link to the source of data.
Admission Tracking System (Database)	ATM	1234	1.0	1/1/2004	1/1/2004	ATM (ATM)	ATM is a web-based application designed to track admission status and provide a link to the source of data.	ATM is a web-based application designed to track admission status and provide a link to the source of data.

- A close approximation to an auditable bandwidth budget
 - You can't manage money without a budget...
 - You can't manage bandwidth without one either...
 - Add to BAC
 - DADMS #
 - Data Transfer Method: SOAP, AJAX, JSON, etc.
 - Data size statistics
 - Replication in use?

28

A pie-graph is not an auditable bandwidth budget.
 I envision more of an excel spreadsheet like the BAC list combined with the NIAPS Application Tracker.
 Tracker would identify: program name, version, BAC, DADMS, POC, Agency, Data Formats, Type of Transfer Method, Protocols, Amount of data transferred, Replication use?

Make an initial assessment of which apps are the heavy BW users. Start with them.

DADMS = DON Application and Database Management Systems



- Capacity alone **does not** improve the Navy's SATCOM network performance.
- **Compression** improves performance of existing bandwidth for **significantly less cost**.
- **Riverbed Steelhead + EXI improves** network performance for structured data (tactical, sensor, administrative) **by greater than 50%**.
- An auditable bandwidth budget is necessary to move forward.
- EXI implemented as an Enterprise Service in CANES expedites EXI to the fleet.



LCDR Steve Debich

Backup Slides




30



Backup Slides: TCP

Slow Start, Congestion Avoidance, Increase Capacity vs
Improve Compression

31



NAVY
POSTGRADUATE
SCHOOL

Capacity, Throughput, and BDP

Capacity alone does not improve SATCOM network performance!

Mbps	Bytes / sec	RTT	BDP	Max Window Size (Bytes)	Throughput
12	1,500,000	0.04	60,000	65,535	13,107,000
8	1,000,000	0.04	40,000	65,535	13,107,000
2	250,000	0.04	10,000	65,535	13,107,000
0.512	64,000	0.04	2,560	65,535	13,107,000
12	1,500,000	0.9	1,350,000	65,535	582,533
8	1,000,000	0.9	900,000	65,535	582,533
2	250,000	0.9	225,000	65,535	582,533
0.512	64,000	0.9	57,600	65,535	582,533

* 0.512 Mbps connection limited by capacity even at 500 ms

- BW-Delay Product (BDP): Bandwidth X Delay
- TCP Window size > BDP; BW limited
- TCP Window size < BDP; RTT limited
 - More than bandwidth alone is needed to improve performance
 - Need to overcome latency
- Most connections never achieve their maximum throughput due to TCP slow start and congestion avoidance.

▶ Pierside connection. Low latency.
Limiting factor is capacity.

▶ SATCOM connection.
High latency.
Limiting factor is latency.

[Return to Capacity vs Throughput](#)

32

BWD = As it's name implies, it is the multiplication of bandwidth (capacity) X delay(round trip time).
 The amount of data that can be sent between ACKs. The amount of data that can be in flight over a link at any given time.

When the TCP window size is greater than the BDP, the path BW(capacity) is the throughput limiting factor.

Look at top part of table

Max throughput for a 40 ms connection is 13 Mbps

It cannot be achieved for any of the channel capacities on the left; therefore the channel is limited by its bandwidth(capacity) and not by the round trip time.

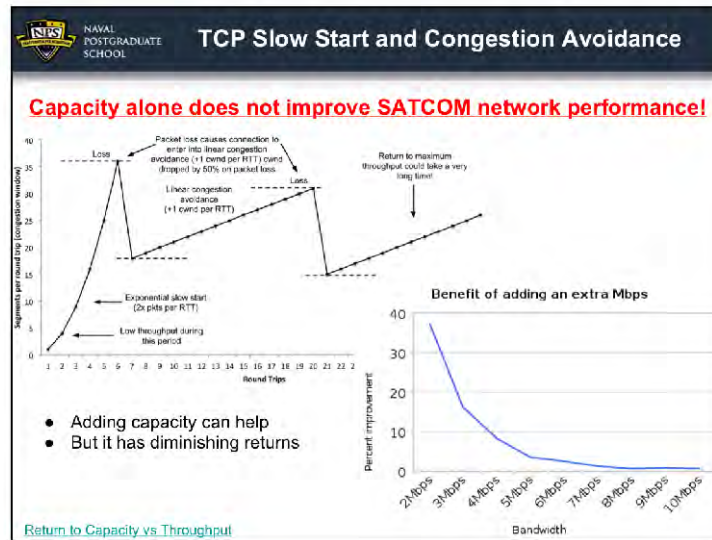
When the TCP window size is less than the BDP, the path round trip time (2x latency) is the throughput limiting factor.

Look at the bottom part of table

Max throughput for a 900 ms connection is 582,533 bps

For an 12, 8, 2 Mbps channel, the channel will never be fully utilized by rather is capped at 582K due to the high latency

- The note identifies that only for the lowest capacity channel 512K does capacity become a limiting factor. In this one case, adding more capacity alone could improve performance.



TCP slow start and congestion avoidance graph retrieved from:
http://www.cisco.com/c/en/us/td/docs/nsite/enterprise/wan/wan_optimization/wan_opt_sg/chap06.html

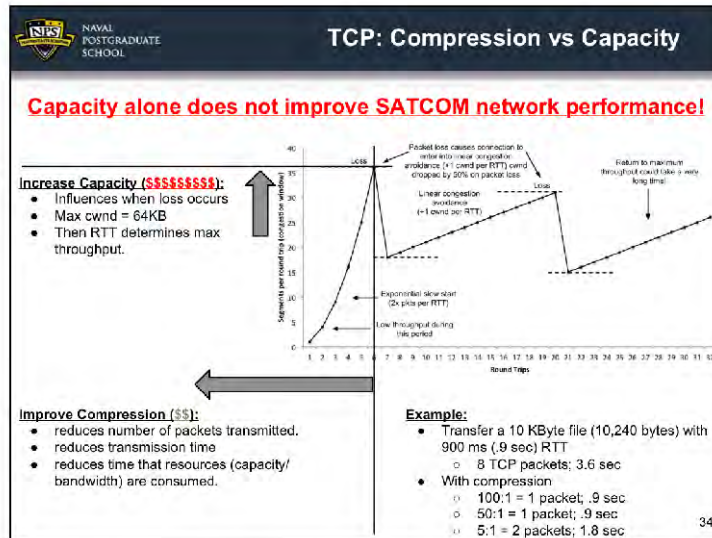
Benefit of adding an extra Mbps graph retrieved from:
<https://docs.google.com/a/chromium.org/viewer?a=v&pid=sites&srcid=Y2hyb21pdW0ub3JnfGRldnxneDoxMzcyOWI1N2I4YzI3NzE2>
 The above link was followed from:
<http://www.chromium.org/spdy>

Mike Belshe is a software engineer who worked on the SPDY Chromium Project which is now the foundation for HTTP 2.0

The information below is copied from:
 Application Acceleration and WAN Optimization Fundamentals by Ted Grevers and Joel Christner

Also
 TCP/IP Guide by Charles M. Kozierok

Slow Start:
 Each side (sender/receiver) is restricted to sending only an amount of data equal to one full-sized segment (Max segment size = 1460 bytes). Ethernet max frame size



Increasing bandwidth/capacity:


- Increasing capacity affects how large the congestion window can grow.
- Max TCP windows size is 64 KB; Determined by TCP header "Window" field = 16 bits, $16^2 = 65,535$
- The TCP maximum throughput equation is then reduced to 64 KB / RTT.
- RTT is then the limiting factor. The largest contributor of RTT for SATCOM is the time it takes to travel to the satellite and back down to earth.

Improving compression:

- Improving compression reduces the number of packets that must be transmitted.
- Reducing the number of packets transmitted reduces the time required for transmission, thereby also reducing the amount of time that resources are being consumed.
- Bandwidth consumption and time to transmit are both reduced.



Research experiment lessons learned



NAVAL
POSTGRADUATE
SCHOOL

Travel, Testing & Lessons Learned

- 17-22 Aug 2014 Initial SPAWAR Visit
 - SPAWAR meet and greet
 - Network Technology and Integration Lab (NTIL).
- 21-23 Oct 2014 Round 1 Testing
 - Lab & Steelhead familiarization
 - Lack of baseline
 - Compression and SDR features not isolated
- 11-13 Nov 2014 Round 2 Testing
 - Adaptive compression - inconclusive
 - HTTP overhead
- 27 - 30 Jan 2015 Round 3 Testing
 - SDR not applied to files 122 bytes and smaller
 - Anomaly reported to Riverbed System Engineer
- 24 Feb - Present
 - Lab distance support – multiple modifications per file

[Return to Testing](#)
36

17-22 Aug 2014

The first trip to SPAWAR was spent talking to various people, mostly set up by Scot. I can provide a list the names later if needed.

Met Eric Otte and saw the lab.

Network Technology and Integration Lab (NTIL).

- Met with various SPAWAR technical leads to better understand the problem space.
- Identified testing capabilities at Network Technology and Integration Lab (NTIL).

21-23 Oct

Round 1


There was a slow start to testing just due to familiarity with the Steelhead configuration and understanding how the Current Connections page worked and displayed data. It was unclear if the LAN/WAN data presented was just payload or if it included protocol headers.

A number of trial runs were needed just to begin to understand how to collect the data.

The results lacked a baseline set of data (baseline is with all features turned off). Without a baseline, it was impossible to calculate performance changes.

My initial plan on how to independently test compression and SDR was wrong. I thought that compression could be tested using a cold transfer of data only.

It was identified that an in-path rule would be needed to isolate compression and SDR. One rule for each. With no in-path rules selected, Compression and SDR was applied to all data

NAVAL POSTGRADUATE SCHOOL

Lessons Learned Summary

- Create a baseline
- Isolate features tested
- Remove all additional overhead
- Thank you OPNAV Studies Program for travel support
- Location of testing data and results listed in thesis appendix

[Return to Testing](#)

37

27-30 Jan

Round 3

- SDR is not actually being applied to files 122 bytes and smaller.
- Anomaly reported to Riverbed System Engineer - Currently under investigation.

HTTP overhead removed by using TCP tools provided by John Schneider
Connection establishment overhead was properly managed and mathematically eliminated by sending a 1 byte “primer” file.

John Schneider identified that it looked like SDR was being “turned off” for files 122 bytes and smaller. The WAN traffic still incurred overhead associated with having SDR on, but repetitive transfers of the same file did not reduce WAN traffic further. The corresponding XML files did have SDR applied properly and were reduced down to 44 bytes. The EXI files however remained above 44 bytes, indicating that SDR was not effecting them.

This unfortunately caused my test set used for testing more than one modification made per file to be voided. The EXI version of the files used were less than 122 bytes and therefore did not have SDR applied to them.

Nick Clouse the Riverbed Federal System Engineer has the question and test data for review and is passing it to others within Riverbed for help identifying the issue.



Compression & Adaptive Compression

38




Compression Results

Compression Ratio Performance Comparison (all values listed are compression ratios)			
File Name	SH Comp	EXI	Performance Improvement
Header	2.06	10.80	5.25
Branch Personnel Type	16.28	26.28	1.61
Menu Meal Recipe Assoc	7.95	17.84	2.24
Planning Transaction Dtl	10.45	27.27	2.61
Target 10	3.05	22.46	7.37
Target 1002-1038	7.49	69.39	9.26
Target 1002-1198	8.92	107.97	12.11
Target 1202-1598	8.94	118.90	13.30
K04.20	3.08	59.67	19.38
K04.24	8.87	56.63	6.38
K04.1	3.13	48.33	15.46

[Return to Compression Results](#)

39

At a minimum EXI compression achieves a 1.61 times greater performance improvement over SH Comp and at most a 19.38 times greater performance improvement over SH Comp.



NAVAL POSTGRADUATE SCHOOL

Adaptive Compression Results

Adaptive Compression Performance Comparison (values listed are compression ratios and performance % difference)

File Name	Gzip	Gzip + SH Adaptive	Performance Difference	EXI	EXI + SH Adaptive	Performance Difference
Header	2.03	1.80	-11%	10.80	7.28	-33%
Branch Personnel Type	16.08	15.70	-2%	26.28	25.19	-4%
Menu Meal Recipe Assoc	8.16	8.16	0%	17.84	17.83	0%
Planning Transation Dtl	10.79	10.79	0%	27.27	27.46	1%
Target 10	3.05	2.90	-5%	22.46	16.47	-27%
Target 1002-1038	7.48	7.39	-1%	69.39	62.15	-10%
Target 1002-1198	8.91	8.88	0%	107.97	104.16	-4%
Target 1202-1598	9.17	9.15	0%	118.90	116.56	-2%
K04 20	3.04	2.90	-5%	59.67	36.28	-39%
K04 24	8.84	8.73	-1%	56.63	52.30	-8%
K04 1	3.09	2.95	-4%	48.33	32.76	-32%

Return to Compression Results

40

EXI experiences a greater performance reduction because the files to be transferred are much smaller to start with. Therefore any overhead added to them by the second attempt to compress, makes them much larger with respect to their compressed size than the Gzip version.

 NAVAL POSTGRADUATE SCHOOL		Backup Results Slides: Cold SDR
Cold SDR		
		41

NAVAL
POSTGRADUATE
SCHOOL

SDR Cold Transfer Results

SDR Cold Transfer Performance Comparison (values listed are compression ratios and performance % difference)

File Name	SH Comp	SH Comp + SDR Cold	Performance Difference	EXI	EXI + SH Comp	EXI + SH Comp + SDR Cold	Performance Difference
Header	2.06	1.78	13%	10.80	7.28	7.28	0%
Branch Personnel Type	16.28	11.64	-28%	25.28	25.29	24.08	-5%
Menu: Meal Recipe Assoc	7.95	6.51	-18%	17.84	18.52	18.07	-2%
Planning Transfers Etl	10.45	9.36	-10%	27.27	28.03	27.36	-2%
Target 10	3.05	2.82	-7%	22.46	16.88	16.88	0%
Target 1002-1038	7.49	6.38	-15%	69.39	62.75	55.92	-11%
Target 1002-1198	8.92	7.81	-12%	107.97	104.50	99.61	-5%
Target 1202-1598	8.91	7.61	-15%	118.90	117.82	113.85	-3%
K04.20	3.08	2.87	-7%	50.67	36.28	36.28	0%
K04.24	8.87	7.67	-14%	56.63	52.67	48.67	-8%
K04.1	3.13	2.92	-6%	48.33	32.76	32.76	0%

42

The comparison view shows the percentage change in performance when applying SDR to first pass or “cold” data. This decrease in overall performance is the “price of admission” or overhead required to reap the benefits of SDR during “warm” transfers.

The EXI data incurs less of a performance decrease due again to its original size. Only now it works in its favor. Since the EXI formatted data is much smaller and compact, the data can fit into less SDR segments and requires less references to be sent to the peer WOC. Also since EXI restructures the data in an optimized way prior to compression, the end result is data that does not have many repeating byte sequences. Therefore there are even less SDR segments.

Note: Header, Target 10, and K04.20 incur no additional overhead. This is an anomaly that Riverbed is looking into. Files 121 bytes and smaller don’t appear to have SDR applied to them. This appears to be a good thing for a cold transfer, but ultimately results in the warm transfer not receiving all the benefits possible for EXI data (because EXI data are the only test files that get below 121 bytes)



Warm SDR

NAVAL
POSTGRADUATE
SCHOOL

SDR Warm Transfer Results

SDR Warm Transfer Performance Comparison (values listed are compression ratios and performance % difference)

File Name	SH Comp	SH Comp + SDR Warm	Performance Improvement	EXI	EXI + SH Comp	EXI + SH Comp + SDR Warm	Performance Improvement
Header	2.06	14.73		7.18	10.80	7.29	1.00
Branch Personnel Type	16.28	277.46		16.78	26.28	25.29	74.00
Naval Medical Recipe Assoc	7.95	235.00		30.08	17.64	18.52	4,175.48
Planning Transition Dtl	10.45	370.18		35.38	22.27	28.03	7,162.23
Target 10	3.05	63.77		20.27	22.46	16.68	16.83
Target 1002-1038	7.49	452.21		60.34	99.39	62.75	596.09
Target 1002-1198	8.92	1,298.02		138.82	107.97	104.50	2,054.35
Target 1202-1598	8.04	325.85		36.44	118.00	117.83	5,909.03
K04-20	3.08	62.02		19.82	39.67	39.26	39.23
K04-24	8.87	385.24		43.54	56.63	52.87	684.70
K04-1	4.14	87.00		71.41	38.11	42.16	57.75

Compression + Cold & Warm SDR Raw Data (all values in bytes)

File Name	XML Size	XML + SH Comp	XML + SH Comp + SDR Cold	XML + SH Comp + SDR Warm	EXI	EXI + SH Comp	EXI + SH Comp + SDR Cold	EXI + SH Comp + SDR Warm
Header	648	335	364	44	60	89	89	89
Branch Personnel Type	26,751	1,640	2,293	98	1,016	1,256	1,109	44
Naval Medical Recipe Assoc	68,756,429	8,605,759	10,561,438	287,805	3,855,746	3,714,662	3,805,590	16,474
Planning Transition Dtl	391,233,416	27,835,966	31,109,097	774,155	10,679,035	10,389,378	10,649,814	40,644
Target 10	2,718	892	963	44	121	161	161	161
Target 1002-1038	76,228	3,500	4,314	58	378	418	469	44
Target 1002-1198	179,992	14,576	16,653	105	1,204	1,244	1,305	44
Target 1202-1598	205,037	23,083	24,192	798	2,117	2,207	2,284	44
K04-20	7,638	877	945	44	65	74	74	74
K04-24	10,177	1,196	1,629	78	512	577	619	44
K04-1	2,948	945	1,008	44	61	90	90	90

[Return to Warm Transfer Results](#)

At a minimum EXI SDR warm transfers achieve 1.32 times greater performance improvement over XML SDR warm transfers and at most 19.05 times greater performance improvement over XML SDR warm transfers.

Warm transfers more than make up for any overhead incurred by the cold transfer or “Price of Admission”

[Eric Otte] is very effective at exploiting and compressing redundant traffic flows. Examples include, two or more users going to the same website, an email attachment being sent to multiple participants, message or track traffic where much of the content is repeated/redundant. These are the big wins we saw in our Fleet trials.
 -Message traffic coming to a ship often has many of the same PLADs listed for the group. This could only be transmitted once, and then followed by an SDR reference.

Single Modification

NAVAL
POSTGRADUATE
SCHOOL

Single Modification

FSM with a single Modification Raw Data (all values in bytes)

File Name	Modification	XML Size	XML + SDR Warm	XML + SH Comp + SDR Warm	EXI + SDR Warm	EXI + SH Comp + SDR Warm
Files Transferred Individually						
Planning Transaction Dtl	Original	291,223,416	N/A	N/A	N/A	N/A
	Beginning	291,223,415	1,950,670	649,317	42,747	43,716
	Middle	291,223,417	1,939,876	654,036	58,084	58,066
	End	291,223,417	1,960,337	617,918	45,068	44,243
Files Transferred Successively						
Planning Transaction Dtl	Original	291,223,416	N/A	N/A	N/A	N/A
	Beginning	291,223,415	1,950,689	625,337	44,807	45,671
	Middle	291,223,417	278,817	280,189	75,799	26,023
	End	291,223,417	315,602	278,724	10,314	10,317

[Return to Single Modification Results](#)

46

Individually

Warm XML Ratios: 445:1 to 471 to 1

Warm EXI Ratios: 5,015:1 to 6,662:1

At a minimum EXI SDR warm transfers achieve 11 times greater performance improvement over XML SDR warm transfers and at most 15 times greater performance improvement over XML SDR warm transfers containing a modification made at the beginning, middle, or end when transferred individually.

-represents sending nearly identical files; two users

Successively

Warm XML Ratios: 466:1 to 1,045:1

Warm EXI Ratios: 6,377:1 to 28,228:1

At a minimum EXI SDR warm transfers achieve 11 times greater performance improvement over XML SDR warm transfers and at most 27 times greater performance improvement over XML SDR warm transfers containing a modification made at the beginning, middle, or end when transferred successively.

-represents sending nearly identical files; three users.



Single Modification

FSM with a single Modification Compression Ratio Comparison				
File Name	Modification	XML + SH Comp + SDR Warm	EXI + SH Comp + SDR Warm	Performance Improvement
Files Transferred Individually				
Planning Transaction Dtl	Original	N/A	N/A	N/A
	Beginning	449	6,662	15
	Middle	445	5,015	11
	End	471	6,582	14
Files Transferred Successively				
Planning Transaction Dtl	Original	N/A	N/A	N/A
	Beginning	466	6,377	14
	Middle	1,039	11,191	11
	End	1,045	28,228	27

[Return to Single Modification Results](#)

47



Single Modification

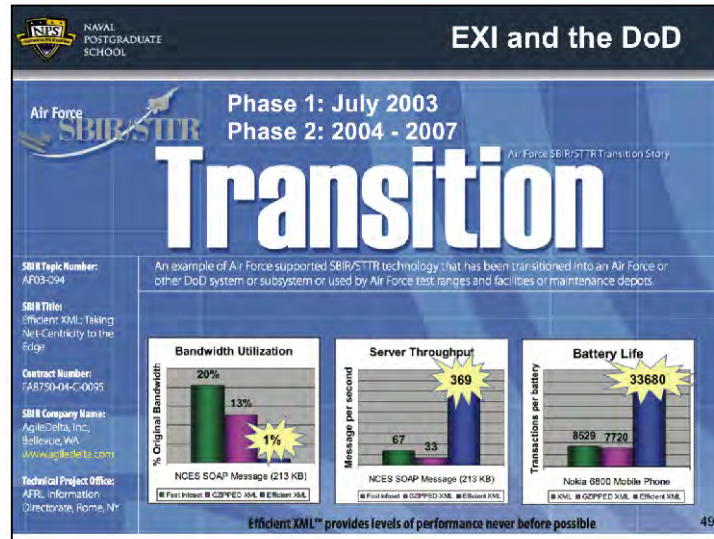
FSM with a single Modification Compression Ratio Comparison				
File Name	Modification	XML + SH Comp + SDR Warm	EXI + SH Comp + SDR Warm	Performance Improvement
Files Transferred Individually				
Planning Transaction Dtl	Original	N/A	N/A	N/A
	Beginning	449	6,662	15
	Middle	445	5,015	11
	End	471	6,582	14
Files Transferred Successively				
Planning Transaction Dtl	Original	N/A	N/A	N/A
	Beginning	466	6,377	14
	Middle	1,039	11,191	11
	End	1,045	28,228	27

[Return to Single Modification Results](#)

47



Graphics for JRAE and JFEX

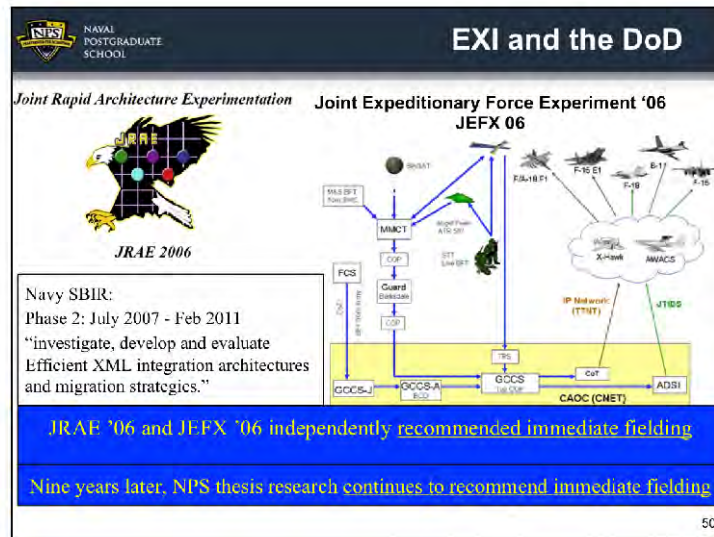


Started as an Air Force SBIR/STTR
Small Business Innovation Research / Small Business Technology Transfer

AF SBIR number: AF03-094
SBIR Title:
Efficient XML: Taking Net-Centricity to the Edge
Company: AgileDelta

Phase 1: Started July 2003
Proposal Title: Extending the Infosphere to Mobile Platforms Using Optimized COTS Technologies
Phase 2: Started August 2004 end September 2007

Abstract:
The objective of this proposal is to demonstrate methods for extending the Battlespace Infosphere to mobile platforms using optimized COTS technologies. The 1999 USAF Scientific Advisory Board report titled Building the Joint Battlespace Infosphere identifies the eXtensible Markup Language (XML) as one of the most promising technologies for representing JBI information objects. Indeed,



Graphics retrieved from individual reports.
Reports are FOUO

Joint Rapid Architecture Experiment '06 (Navy)

The 2006 Joint Rapid Architecture Experiment (JRAE 06) examined the ability of the proposed Joint Targeting Service Oriented Architecture (SOA) to improve the timeliness and accuracy of planned and emergent Joint strike operations. JRAE 06 was conducted in August 2006 at the SPAWAR Advanced Concepts Laboratory, San Diego, California. JRAE 06 focused on improving the flow of tactical information to and from tactical edge users, enabling the participation of coalition partners in the US Joint Target Management (JTM) SOA, and applying Information Assurance (IA) protocols to the JTM service

Joint Expeditionary Force Experiment '06 (Air Force)

Navy SBIR

Phase 2: Start July 2007 end Feb 2011

Topic: AF03-094 (carried over from the Air Force)

Abstract:

As the DoD shifts toward Network-Centric operations, the vision of sharing common information objects between command centers, aircraft, ships and ground forces over a global network seems closer than ever. One of the

LIST OF REFERENCES

- Air Force Research Laboratory. (2003). Extending the infosphere to mobile platforms using optimized COTS technologies. Retrieved February 15, 2015, from <http://www.afsbirsttr.com/award/AwardDetails.aspx?pk=9302>
- The Apache Software Foundation. (2015). Apache module mod_deflate. Retrieved January 15, 2015, from http://httpd.apache.org/docs/2.4/mod/mod_deflate.html
- Belshe, M. (2010, April). More bandwidth doesn't matter (much). [Google Doc]. Retrieved from <https://docs.google.com/a/chromium.org/viewer?a=v&pid=sites&srcid=Y2hyb21pdW0ub3JnfGRldnxneDoxMzcyOWI1N2I4YzI3NzE2>
- Belshe, M., Thomson, M., & Peon, R. (2014). Hypertext transfer protocol version 2. Retrieved from <https://tools.ietf.org/html/draft-ietf-httpbis-http2-16>
- Bentrup, J., Otte, E., Chan, B., Vavrich, D., & Gingras, D. (2012). *At-sea testing of a wide area network optimization device (U)*. CNA Corporation. DRM-2012-U-002246-Final. Retrieved from <https://www.dtic.mil/DTICOnline/downloadPdf.search?collectionId=tr&docId=ADB383695>
- Betts, B. (2007, March). SSL traffic clogs wans. *PCWorld*. Retrieved from <http://www.pcworld.com/article/129648/article.html>
- Blelloch, G. E. (2013). *Introduction to data compression*. Retrieved from <http://www.cs.cmu.edu/afs/cs/project/pscico-guyb/realworld/www/compression.pdf>
- Boeing. (n.d.). *Transformational wideband communication capabilities for the warfighter*. [Fact sheet]. Retrieved from http://www.boeing.com/boeing/defense-space/space/bss/factsheets/702/wgs/wgs_factsheet.page
- Boeing. (2013). Wideband global SATCOM (WGS) backgrounder. Retrieved from http://www.boeing.com/assets/pdf/defense-space/space/bss/factsheets/702/wgs/docs/Bkgd_WGS_1013.pdf
- Bos, B. (2001). XML in 10 points. Retrieved April 7, 2014, from <http://www.w3.org/XML/1999/XML-in-10-points.html.en>
- Bournez, C. (2009). Efficient XML interchange evaluation. Retrieved April 14, 2014, from <http://www.w3.org/TR/2009/WD-exi-evaluation-20090407/#compactness-results>

- Bray, T., Paoli, J., & Sperberg-McQueen, C. M. (1998). Extensible markup language (xml) 1.0: W3c recommendation 10-february-1998. Retrieved April 14, 2014, from <http://www.w3.org/TR/1998/REC-xml-19980210>
- Chief of Naval Operations for Information Dominance. (2013). Navy strategy for achieving information dominance, 2013-2017. Washington, DC: Author. Retrieved from <http://www.dtic.mil/docs/citations/ADA571217>
- The Chromium Projects. (2014). SPDY: An experimental protocol for a faster web. Retrieved May 30, 2014, from <http://www.chromium.org/spdy/spdy-whitepaper>
- Clark, J., & Van Oorschot, P. C. (2013). SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. In *2013 IEEE Symposium on Security and Privacy* (pp. 511–525). IEEE. doi:10.1109/SP.2013.41
- Cokus, M., & Pericas-Geertsen, S. (2005). Xml binary characterization properties. Retrieved from <http://www.w3.org/TR/2005/NOTE-xbc-properties-20050331/>
- Couch, L. W. (2013). *Digital & Analog Communication Systems* (8th ed.). Upper Saddle River, NJ: Pearson Education.
- Curran, J. (2014). Snowden leaks said to spur encryption by internet firms. *Cyber Security Policy Report*. Retrieved from <http://search.proquest.com/docview/1521287619?accountid=12702>
- Daemen, J., & Rijmen, V. (Eds.). (2002). *Fast Software Encryption* (Vol. 2365). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/3-540-45661-9
- Debich, S., Hill, B., Miller, S., & Brutzman, D. (2014). Being efficient with bandwidth. *United States Naval Institute Proceedings*, 140(7), 76–78. Retrieved from <http://www.usni.org/magazines/proceedings/2014-07/professional-notes>
- Deng, Y., & Manoharan, S. (2013). A review of network latency optimization techniques. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (pp. 20–25). IEEE. doi:10.1109/PACRIM.2013.6625442
- Department of the Air Force. (2012). U.S. Air Force factsheet: Wideband global SATCOM satellite. [Fact sheet]. Retrieved December 23, 2014, from <http://www.afspc.af.mil/library/factsheets/factsheet.asp?id=5582>
- Deutsch, L. P. (1996). DEFLATE compressed data format specification version 1.3. Retrieved May 31, 2014, from <http://tools.ietf.org/html/rfc1951>
- DOD IT Standards Registry. (2011). DISR W3C EXI format 1.0. Retrieved December 20, 2014, from <https://gtg.csd.disa.mil/dISR/standards/view.html?id=300573>

- Fall, K. (2003). A delay-tolerant network architecture for challenged Internets. *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications - SIGCOMM '03*, 27–34. doi:10.1145/863956.863960
- Fielding, R., & Reschke, J. (2014). Hypertext transfer protocol (HTTP/1.1): Semantics and content. Retrieved December 14, 2014, from <https://tools.ietf.org/html/rfc7231>
- Goldman, O., & Lenkov, D. (2005). XML binary characterization. Retrieved from <http://www.w3.org/TR/xbc-characterization/>
- Grevers, T. J., & Christner, J. (2008). *Application acceleration and WAN optimization fundamentals*. [Safari Books Online version]. Retrieved from <http://techbus.safaribooksonline.com/>
- Grigorik, I. (2012). Latency: The new web performance bottleneck. Retrieved May 30, 2014, from <https://www.igvita.com/2012/07/19/latency-the-new-web-performance-bottleneck/>
- Grigorik, I. (2013). *High performance browser networking: What every web developer should know about networking and web performance*. [Safari Books Online version]. Retrieved from <http://techbus.safaribooksonline.com/>
- Henderson, T. R., & Katz, R. H. (1999). *TCP performance over satellite channels*. University of California Berkeley. Retrieved from <http://www.eecs.berkeley.edu/Pubs/TechRpts/1999/CSD-99-1083.pdf>
- Hill, B. W. (2015). *Evaluation of efficient XML interchange for large datasets and as alternative to binary JSON encodings*. Monterey, CA: Naval Postgraduate School.
- Hollenbeck, S. (2004). TLS compression methods. Retrieved August 26, 2014, from <http://www.ietf.org/rfc/rfc3749.txt>
- Intel. (2013). *Upsurge in encrypted traffic drives demand for cost-efficient SSL application delivery [White Paper]*. Retrieved from <http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/cost-efficient-ssl-application-delivery-paper.pdf>
- International Organization for Standardization. (2007). ISO/IEC 24824-1:2007: Information technology - generic applications of asn.1: Fast infoset. Geneva, Switzerland: ISO/IEC. Retrieved from <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>
- Internet Assigned Numbers Authority. (2014). Hypertext transfer protocol (HTTP) parameters. Retrieved January 15, 2015, from <http://www.iana.org/assignments/http-parameters/http-parameters.xhtml>

- Jacobson, V., Braden, R., & Borman, D. (1992). TCP extensions for high performance. Retrieved February 6, 2015, from <https://tools.ietf.org/html/rfc1323>
- Kangasharju, J. (2008). *XML messaging for mobile devices*. University of Helsinki Finland. Retrieved from <http://helda.helsinki.fi/handle/10138/21347>
- Kattan, A. (2010). Universal intelligent data compression systems: A review. *2010 2nd Computer Science and Electronic Engineering Conference (CEECE)*. doi:10.1109/CEECE.2010.5606482
- Kay, R. (2009). Pragmatic network latency engineering fundamental facts and analysis. [PDF]. Retrieved from http://nhne-pulse.org/wp-content/uploads/2011/01/Introduction_to_Network_Latency_Engineering.pdf
- Kozierok, C. M. (2005). *The TCP/IP guide: A comprehensive, illustrated Internet protocols reference*. [Safari Books Online version]. Retrieved from <http://techbus.safaribooksonline.com/>
- Kumar, R., Taggart, D., Monzingo, R., & Goo, G. (2005). Wideband gapfiller satellite (WGS) system. In *2005 IEEE Aerospace Conference* (pp. 1410–1417). IEEE. doi:10.1109/AERO.2005.1559431
- Kyusakov, R. (2014). *Efficient web services for end-to-end interoperability of embedded systems*. Luleå University of Technology. Retrieved from http://pure.ltu.se/portal/files/100108844/Rumen_Kyusakov.pdf
- Le Hegaret, P. (2005). Charter of the efficient xml interchange working group. Retrieved November 8, 2014, from <http://www.w3.org/2005/09/exi-charter-final.html>
- Liefke, H., & Suciu, D. (2000). XMill: An efficient compressor for xml data. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (pp. 153–164). doi:10.1145/342009.335405
- Martin, B., & Jano, B. (1999). WAP binary XML content format. Retrieved January 7, 2015, from <http://www.w3.org/1999/06/NOTE-wbxml-19990624>
- McMillan, T. (2012). *Cisco networking essentials*. [Safari Books Online version]. Retrieved from <http://techbus.safaribooksonline.com/>
- Meyer, C., & Schwenk, J. (2013). Lessons learned from previous ssl/tls attacks: A brief chronology of attacks and weaknesses. *IACR Cryptology ePrint Archive, 2013*, 49. Retrieved from International Association for Cryptologic Research ePrint website: <https://eprint.iacr.org/2013/049.pdf>
- Microsoft. (2015). HTTP compression <httpCompression>. Retrieved November 8, 2014, from <http://www.iis.net/configreference/system.webserver/httpcompression>

- Morgan, B., & Dennis, C. (2003). *CCNP BCRAN self-study: Managing network performance with queuing and compression* (2nd ed.). Cisco Press. Retrieved from <http://www.ciscopress.com/articles/article.asp?p=102233&seqNum=6>
- Morse, K. G. (2005). Compression tools compared. *Linux Journal*, (137). Retrieved from <http://www.linuxjournal.com/node/8051/print>
- Navy SBIR. (2007). Extending the infosphere to mobile platforms using optimized COTS technologies. Retrieved February 15, 2015, from <http://www.navysbirprogram.com/NavySearch/Award/award.aspx?pk=C52705C7-4388-493A-BC4F-C1EE5973598E>
- Nirmala, M. B. (2014). WAN optimization tools, techniques and research issues for cloud-based big data analytics (pp. 280–285). doi:10.1109/WCCCT.2014.72
- Oxford University Press. (2013). About. Retrieved February 10, 2015, from <http://public.oed.com/about/>
- Peintner, D., Kosch, H., & Heuer, J. (2009). Efficient xml interchange for rich internet applications. In *2009 IEEE International Conference on Multimedia and Expo* (pp. 149–152). IEEE. doi:10.1109/ICME.2009.5202458
- Peintner, D., & Pericas-Geertsen, S. (2007). Efficient XML interchange primer. Retrieved from <http://www.w3.org/TR/2007/WD-exi-primer-20071219>
- Porche, I. R., Wilson, B., Johnson, E.-E., Tierney, S., & Saltzman, E. (2014). Data flood: Helping the navy address the rising tide of sensor information. Santa Monica, CA: RAND National Defense Research Institute. Retrieved from http://www.rand.org/pubs/research_reports/RR315.html
- Riverbed Technology. (2013a). Steelhead appliance deployment guide. Retrieved from <https://support.riverbed.com/content/support/software/steelhead/cx-appliance.html>
- Riverbed Technology. (2013b). Steelhead management console user's guide. Retrieved from <https://support.riverbed.com/content/support/software/steelhead/cx-appliance.html>
- Saha, S., Jamtgaard, M., & Villasenor, J. (2001). Bringing the wireless internet to mobile devices. *Computer*, 34(6), 54–58. doi:10.1109/2.928622
- Sakr, S. (2009). XML compression techniques: A survey and comparison. *Journal of Computer and System Sciences*, 75(5), 303–322. doi:10.1016/j.jcss.2009.01.004
- Salomon, D. (2008). *A concise introduction to data compression*. London: Springer-Verlag.

- Schneider, J. (2008). *Efficient XML taking net-centric operations to the edge*. Retrieved from http://dodccrp.org/events/13th_icrts_2008/CD/html/papers/167.pdf
- Schneider, J., Kamiya, T., Peintner, D., & Kyusakov, R. (2014). Efficient xml interchange (exi) format 1.0 (second edition). Retrieved from <http://www.w3.org/TR/2014/REC-exi-20140211/>
- Skorupa, J., & Munch, B. (2014). Magic quadrant for wan optimization. Retrieved from <http://www.gartner.com/technology/reprints.do?id=1-1SCJEST&ct=140328&st=sb>
- Snyder, S., McGregor, D., & Brutzman, D. (2009). *Efficient xml interchange: Compact, efficient, and standards-based xml for modeling and simulation*. Naval Postgraduate School. Retrieved from <http://hdl.handle.net/10945/30774>
- Thompson, S. (2004). The Value of Compression for Data Protection Over TCP/IP WANs. *Computer Technology Review*, 24(6), 2.
- Tierney, B. L. (2008). *Bulk data transfer techniques for high-speed wide-area networks*. [PowerPoint slides]. Retrieved from Internet2 website: <http://www.internet2.edu/presentations/jt2008jul/20080721-tierney.pdf>
- W3C. (2014). Efficient xml interchange working group. Retrieved June 1, 2014, from <http://www.w3.org/XML/EXI/>
- The WAP Forum. (2000). Wireless application protocol white paper: Wireless internet today. Retrieved from http://www.wapforum.org/what/WAP_white_pages.pdf
- Zhang, Y., Ansari, N., Wu, M., & Yu, H. (2012). On wide area network optimization. *IEEE Communications Surveys & Tutorials*, 14(4), 1090–1113. doi:10.1109/SURV.2011.092311.00071

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California